

Optimizations

Do no harm, i.e., do not change the meaning of the program in any way

Changing a 3 to a 4 is obviously bad, but replacing error-signalling code with code that does not signal the error *is* harm.

Optimizations

Consider a *<insert life-critical device with software here>* that catches such errors and goes into a fallback, safe state

- When the pacemaker's software has an error; catch the error, stop stimulating the heart, and call 911
- When the autopilot software has an error; catch the error and wake up the pilot
- When the train crossing software crashes, catch the error and block all cars from passing

Consider this example L3 program

```

((let ((arr (new-array 10 3)))
  (let ((_before (print arr)))
    (let ((_inc (:inc arr 0 4)))
      (print arr))))
(:inc
 (a i y)
 (let ((len (alen a)))
  (let ((altlen (< i len)))
   (if altlen
    (let ((aval (aref a i)))
      (let ((newaval (+ aval y)))
        (let ((_set (aset a i newaval)))
          (let ((nexti (+ i 1)))
            (:inc a nexti y))))))
    0))))))

```

Here is its compilation into L2
(note how `main` is compiled)

(call :L 0)

```
:L_0
(eax <- (allocate 21 7))
(arr <- eax)
(eax <- (print arr))
(_before <- eax)
(ecx <- arr)
(edx <- 1)
(eax <- 9)
(call :inc)
(_inc <- eax)
(eax <- (print arr))
(eax <- eax)
(return)
```

```
:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <<= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <<= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <<= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 += a)
((mem x_3 4) <- newaval)
(_set <- 1)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(tail-call :inc)
```

We are going to change the `tail-call` into a `goto` because it is safe here; later we'll see how to do this in general

```

                                :inc
                                (a <- ecx)
                                (i <- edx)
                                (y <- eax)
                                (len <- (mem a 0))
                                (len <= 1)
                                (len += 1)
                                (altlen <- i < len)
                                (altlen += altlen)
                                (altlen += 1)
                                (cjump altlen = 1 :L_2 :L_1)
                                :L_2
                                (eax <- 1)
                                (return)
                                :L_1
                                (x_1 <- i)
                                (bnds_2 <- (mem a 0))
                                (bnds_2 <= 1)
                                (bnds_2 += 1)
                                (cjump x_1 < bnds_2 :oka_4 :die_3)
                                :oka_4
                                (cjump 0 <= x_1 :okb_5 :die_3)
                                :die_3
                                (eax <- (array-error a i))
                                :okb_5
                                (x_1 >>= 1)
                                (x_1 *= 4)
                                (x_1 += a)
                                (aval <- (mem x_1 4))
                                (newaval <- aval)
                                (newaval += y)
                                (newaval -= 1)
                                (x_3 <- i)
                                (bnds_4 <- (mem a 0))
                                (bnds_4 <= 1)
                                (bnds_4 += 1)
                                (cjump x_3 < bnds_4 :oka_7 :die_6)
                                :oka_7
                                (cjump 0 <= x_3 :okb_8 :die_6)
                                :die_6
                                (eax <- (array-error a i))
                                :okb_8
                                (x_3 >>= 1)
                                (x_3 *= 4)
                                (x_3 += a)
                                ((mem x_3 4) <- newaval)
                                (_set <- 1)
                                (nexti <- i)
                                (nexti += 2)
                                (ecx <- a)
                                (edx <- nexti)
                                (eax <- y)
                                (goto :inc)

(:inc
 (a i y)
 (let ((len (alen a)))
   (let ((altlen (< i len)))
     (if altlen
       (let ((aval (aref a i))
             (let ((newaval (+ aval y)))
               (let ((_set (aset a i newaval)))
                 (let ((nexti (+ i 1)))
                   (:inc a nexti y))))))
       0))))

```


Analysis \Rightarrow Optimizatoin

Run an analysis to determine some information about the program

Use that information to transform the program into (hopefully) a better one

Dead code elimination

We can use liveness analysis to identify useless code

Remove any instruction that assigns to a variable, if the variable is not in the out set of that instruction

	in	out
1: :inc	0	0
2: (a <- ecx)	0	0
3: (i <- edx)	0	0
4: (y <- eax)	0	0
5: (len <- (mem a 0))	0	0
6: (len <<= 1)	0	0
7: (len += 1)	0	0
8: (altlen <- i < len)	0	0
9: (altlen += altlen)	0	0
10: (altlen += 1)	0	0
11: (cjump altlen = 1 :L_2 :L_1)	0	0
12: :L_2	0	0
13: (eax <- 1)	0	0
14: (return)	0	0
15: :L_1	0	0
16: (x_1 <- i)	0	0
17: (bnds_2 <- (mem a 0))	0	0
18: (bnds_2 <<= 1)	0	0
19: (bnds_2 += 1)	0	0
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	0	0
21: :oka_4	0	0
22: (cjump 0 <= x_1 :okb_5 :die_3)	0	0
23: :die_3	0	0
24: (eax <- (array-error a i))	0	0
25: :okb_5	0	0
26: (x_1 >>= 1)	0	0
27: (x_1 *= 4)	0	0
28: (x_1 += a)	0	0
29: (aval <- (mem x_1 4))	0	0
30: (newaval <- aval)	0	0
31: (newaval += y)	0	0
32: (newaval -= 1)	0	0
33: (x_3 <- i)	0	0
34: (bnds_4 <- (mem a 0))	0	0
35: (bnds_4 <<= 1)	0	0
36: (bnds_4 += 1)	0	0
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	0	0
38: :oka_7	0	0
39: (cjump 0 <= x_3 :okb_8 :die_6)	0	0
40: :die_6	0	0
41: (eax <- (array-error a i))	0	0
42: :okb_8	0	0
43: (x_3 >>= 1)	0	0
44: (x_3 *= 4)	0	0
45: (x_3 += a)	0	0
46: ((mem x_3 4) <- newaval)	0	0
47: (_set <- 1)	0	0
48: (nexti <- i)	0	0
49: (nexti += 2)	0	0
50: (ecx <- a)	0	0
51: (edx <- nexti)	0	0

	in	out
1: :inc	()	0
2: (a <- ecx)	(ecx)	0
3: (i <- edx)	(edx)	0
4: (y <- eax)	(eax)	0
5: (len <- (mem a 0))	(a)	0
6: (len <=<= 1)	(len)	0
7: (len += 1)	(len)	0
8: (altlen <- i < len)	(i len)	0
9: (altlen += altlen)	(altlen)	0
10: (altlen += 1)	(altlen)	0
11: (cjump altlen = 1 :L_2 :L_1)	(altlen)	0
12: :L_2	()	0
13: (eax <- 1)	()	0
14: (return)	(ebx edi esi)	0
15: :L_1	()	0
16: (x_1 <- i)	(i)	0
17: (bnds_2 <- (mem a 0))	(a)	0
18: (bnds_2 <=<= 1)	(bnds_2)	0
19: (bnds_2 += 1)	(bnds_2)	0
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(bnds_2 x_1)	0
21: :oka_4	()	0
22: (cjump 0 <= x_1 :okb_5 :die_3)	(x_1)	0
23: :die_3	()	0
24: (eax <- (array-error a i))	(a i)	0
25: :okb_5	()	0
26: (x_1 >>= 1)	(x_1)	0
27: (x_1 *= 4)	(x_1)	0
28: (x_1 += a)	(a x_1)	0
29: (aval <- (mem x_1 4))	(x_1)	0
30: (newaval <- aval)	(aval)	0
31: (newaval += y)	(newaval y)	0
32: (newaval -= 1)	(newaval)	0
33: (x_3 <- i)	(i)	0
34: (bnds_4 <- (mem a 0))	(a)	0
35: (bnds_4 <=<= 1)	(bnds_4)	0
36: (bnds_4 += 1)	(bnds_4)	0
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(bnds_4 x_3)	0
38: :oka_7	()	0
39: (cjump 0 <= x_3 :okb_8 :die_6)	(x_3)	0
40: :die_6	()	0
41: (eax <- (array-error a i))	(a i)	0
42: :okb_8	()	0
43: (x_3 >>= 1)	(x_3)	0
44: (x_3 *= 4)	(x_3)	0
45: (x_3 += a)	(a x_3)	0
46: ((mem x_3 4) <- newaval)	(newaval x_3)	0
47: (_set <- 1)	()	0
48: (nexti <- i)	(i)	0
49: (nexti += 2)	(nexti)	0
50: (ecx <- a)	(a)	0
51: (edx <- nexti)	(nexti)	0

	in	out
1: :inc	()	(ecx)
2: (a <- ecx)	(ecx)	(edx)
3: (i <- edx)	(edx)	(eax)
4: (y <- eax)	(eax)	(a)
5: (len <- (mem a 0))	(a)	(len)
6: (len <=<= 1)	(len)	(len)
7: (len += 1)	(len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen)	(altlen)
11: (cjump altlen = 1 :L_2 :L_1)	(altlen)	()
12: :L_2	()	()
13: (eax <- 1)	()	(ebx edi esi)
14: (return)	(ebx edi esi)	()
15: :L_1	()	(i)
16: (x_1 <- i)	(i)	(a)
17: (bnds_2 <- (mem a 0))	(a)	(bnds_2)
18: (bnds_2 <=<= 1)	(bnds_2)	(bnds_2)
19: (bnds_2 += 1)	(bnds_2)	(bnds_2 x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(bnds_2 x_1)	()
21: :oka_4	()	(x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(x_1)	()
23: :die_3	()	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	()	(x_1)
26: (x_1 >>= 1)	(x_1)	(x_1)
27: (x_1 *= 4)	(x_1)	(a x_1)
28: (x_1 += a)	(a x_1)	(x_1)
29: (aval <- (mem x_1 4))	(x_1)	(aval)
30: (newaval <- aval)	(aval)	(newaval y)
31: (newaval += y)	(newaval y)	(newaval)
32: (newaval -= 1)	(newaval)	(i)
33: (x_3 <- i)	(i)	(a)
34: (bnds_4 <- (mem a 0))	(a)	(bnds_4)
35: (bnds_4 <=<= 1)	(bnds_4)	(bnds_4)
36: (bnds_4 += 1)	(bnds_4)	(bnds_4 x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(bnds_4 x_3)	()
38: :oka_7	()	(x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(x_3)	()
40: :die_6	()	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	()	(x_3)
43: (x_3 >>= 1)	(x_3)	(x_3)
44: (x_3 *= 4)	(x_3)	(a x_3)
45: (x_3 += a)	(a x_3)	(newaval x_3)
46: ((mem x_3 4) <- newaval)	(newaval x_3)	()
47: (_set <- 1)	()	(i)
48: (nexti <- i)	(i)	(nexti)
49: (nexti += 2)	(nexti)	(a)
50: (ecx <- a)	(a)	(nexti)
51: (edx <- nexti)	(nexti)	(y)

	in	out
1: :inc	(ecx)	(ecx)
2: (a <- ecx)	(ecx edx)	(edx)
3: (i <- edx)	(eax edx)	(eax)
4: (y <- eax)	(a eax)	(a)
5: (len <- (mem a 0))	(a)	(len)
6: (len <=<= 1)	(len)	(len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen)	(altlen)
11: (cjump altlen = 1 :L_2 :L_1)	(altlen)	()
12: :L_2	()	()
13: (eax <- 1)	(ebx edi esi)	(ebx edi esi)
14: (return)	(ebx edi esi)	()
15: :L_1	(i)	(i)
16: (x_1 <- i)	(a i)	(a)
17: (bnds_2 <- (mem a 0))	(a)	(bnds_2)
18: (bnds_2 <=<= 1)	(bnds_2)	(bnds_2)
19: (bnds_2 += 1)	(bnds_2 x_1)	(bnds_2 x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(bnds_2 x_1)	()
21: :oka_4	(x_1)	(x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(x_1)	()
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(x_1)	(x_1)
26: (x_1 >>= 1)	(x_1)	(x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1)
28: (x_1 += a)	(a x_1)	(x_1)
29: (aval <- (mem x_1 4))	(x_1)	(aval)
30: (newaval <- aval)	(aval y)	(newaval y)
31: (newaval += y)	(newaval y)	(newaval)
32: (newaval -= 1)	(i newaval)	(i)
33: (x_3 <- i)	(a i)	(a)
34: (bnds_4 <- (mem a 0))	(a)	(bnds_4)
35: (bnds_4 <=<= 1)	(bnds_4)	(bnds_4)
36: (bnds_4 += 1)	(bnds_4 x_3)	(bnds_4 x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(bnds_4 x_3)	()
38: :oka_7	(x_3)	(x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(x_3)	()
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(x_3)	(x_3)
43: (x_3 >>= 1)	(x_3)	(x_3)
44: (x_3 *= 4)	(a x_3)	(a x_3)
45: (x_3 += a)	(a newaval x_3)	(newaval x_3)
46: ((mem x_3 4) <- newaval)	(newaval x_3)	()
47: (_set <- 1)	(i)	(i)
48: (nexti <- i)	(i)	(nexti)
49: (nexti += 2)	(a nexti)	(a)
50: (ecx <- a)	(a nexti)	(nexti)
51: (edx <- nexti)	(nexti y)	(y)

	in	out
1: :inc	(ecx)	(ecx edx)
2: (a <- ecx)	(ecx edx)	(eax edx)
3: (i <- edx)	(eax edx)	(a eax)
4: (y <- eax)	(a eax)	(a)
5: (len <- (mem a 0))	(a)	(len)
6: (len <=<= 1)	(len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen)	(altlen)
11: (cjump altlen = 1 :L_2 :L_1)	(altlen)	(i)
12: :L_2	()	(ebx edi esi)
13: (eax <- 1)	(ebx edi esi)	(ebx edi esi)
14: (return)	(ebx edi esi)	(i)
15: :L_1	(i)	(a i)
16: (x_1 <- i)	(a i)	(a)
17: (bnds_2 <- (mem a 0))	(a)	(bnds_2)
18: (bnds_2 <=<= 1)	(bnds_2)	(bnds_2 x_1)
19: (bnds_2 += 1)	(bnds_2 x_1)	(bnds_2 x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(bnds_2 x_1)	(a i x_1)
21: :oka_4	(x_1)	(x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(x_1)	(x_1)
26: (x_1 >>= 1)	(x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1)
28: (x_1 += a)	(a x_1)	(x_1)
29: (aval <- (mem x_1 4))	(x_1)	(aval y)
30: (newaval <- aval)	(aval y)	(newaval y)
31: (newaval += y)	(newaval y)	(i newaval)
32: (newaval -= 1)	(i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a)
34: (bnds_4 <- (mem a 0))	(a)	(bnds_4)
35: (bnds_4 <=<= 1)	(bnds_4)	(bnds_4 x_3)
36: (bnds_4 += 1)	(bnds_4 x_3)	(bnds_4 x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(bnds_4 x_3)	(a i x_3)
38: :oka_7	(x_3)	(x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(x_3)	(x_3)
43: (x_3 >>= 1)	(x_3)	(a x_3)
44: (x_3 *= 4)	(a x_3)	(a newaval x_3)
45: (x_3 += a)	(a newaval x_3)	(newaval x_3)
46: ((mem x_3 4) <- newaval)	(newaval x_3)	(i)
47: (_set <- 1)	(i)	(i)
48: (nexti <- i)	(i)	(a nexti)
49: (nexti += 2)	(a nexti)	(a nexti)
50: (ecx <- a)	(a nexti)	(nexti y)
51: (edx <- nexti)	(nexti y)	(y)

	in	out
1: :inc	(ecx edx)	(ecx edx)
2: (a <- ecx)	(eax ecx edx)	(eax edx)
3: (i <- edx)	(a eax edx)	(a eax)
4: (y <- eax)	(a eax)	(a)
5: (len <- (mem a 0))	(a)	(len)
6: (len <=<= 1)	(i len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen)	(altlen)
11: (cjump altlen = 1 :L_2 :L_1)	(altlen i)	(i)
12: :L_2	(ebx edi esi)	(ebx edi esi)
13: (eax <- 1)	(ebx edi esi)	(ebx edi esi)
14: (return)	(ebx edi esi i)	(i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a)
17: (bnds_2 <- (mem a 0))	(a)	(bnds_2)
18: (bnds_2 <=<= 1)	(bnds_2 x_1)	(bnds_2 x_1)
19: (bnds_2 += 1)	(bnds_2 x_1)	(bnds_2 x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(x_1)	(x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(x_1)	(x_1)
26: (x_1 >>= 1)	(a x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1)
28: (x_1 += a)	(a x_1)	(x_1)
29: (aval <- (mem x_1 4))	(x_1 y)	(aval y)
30: (newaval <- aval)	(aval y)	(newaval y)
31: (newaval += y)	(i newaval y)	(i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a)
34: (bnds_4 <- (mem a 0))	(a)	(bnds_4)
35: (bnds_4 <=<= 1)	(bnds_4 x_3)	(bnds_4 x_3)
36: (bnds_4 += 1)	(bnds_4 x_3)	(bnds_4 x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i x_3)	(a i x_3)
38: :oka_7	(x_3)	(x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(x_3)	(x_3)
43: (x_3 >>= 1)	(a x_3)	(a x_3)
44: (x_3 *= 4)	(a newaval x_3)	(a newaval x_3)
45: (x_3 += a)	(a newaval x_3)	(newaval x_3)
46: ((mem x_3 4) <- newaval)	(i newaval x_3)	(i)
47: (_set <- 1)	(i)	(i)
48: (nexti <- i)	(a i)	(a nexti)
49: (nexti += 2)	(a nexti)	(a nexti)
50: (ecx <- a)	(a nexti y)	(nexti y)
51: (edx <- nexti)	(nexti y)	(y)

	in	out
1: :inc	(ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax)
4: (y <- eax)	(a eax)	(a)
5: (len <- (mem a 0))	(a)	(i len)
6: (len <=<= 1)	(i len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen)	(altlen i)
11: (cjump altlen = 1 :L_2 :L_1)	(altlen i)	(a ebx edi esi i)
12: :L_2	(ebx edi esi)	(ebx edi esi)
13: (eax <- 1)	(ebx edi esi)	(ebx edi esi i)
14: (return)	(ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a)
17: (bnds_2 <- (mem a 0))	(a)	(bnds_2 x_1)
18: (bnds_2 <=<= 1)	(bnds_2 x_1)	(bnds_2 x_1)
19: (bnds_2 += 1)	(bnds_2 x_1)	(a bnds_2 i x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(x_1)	(a i x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(x_1)	(a x_1)
26: (x_1 >>= 1)	(a x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1)
28: (x_1 += a)	(a x_1)	(x_1 y)
29: (aval <- (mem x_1 4))	(x_1 y)	(aval y)
30: (newaval <- aval)	(aval y)	(i newaval y)
31: (newaval += y)	(i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a)
34: (bnds_4 <- (mem a 0))	(a)	(bnds_4 x_3)
35: (bnds_4 <=<= 1)	(bnds_4 x_3)	(bnds_4 x_3)
36: (bnds_4 += 1)	(bnds_4 x_3)	(a bnds_4 i x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i x_3)	(a i x_3)
38: :oka_7	(x_3)	(a i x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(x_3)	(a x_3)
43: (x_3 >>= 1)	(a x_3)	(a newaval x_3)
44: (x_3 *= 4)	(a newaval x_3)	(a newaval x_3)
45: (x_3 += a)	(a newaval x_3)	(i newaval x_3)
46: ((mem x_3 4) <- newaval)	(i newaval x_3)	(i)
47: (_set <- 1)	(i)	(a i)
48: (nexti <- i)	(a i)	(a nexti)
49: (nexti += 2)	(a nexti)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(nexti y)
51: (edx <- nexti)	(nexti y)	(y)

	in	out
1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax)
4: (y <- eax)	(a eax)	(a)
5: (len <- (mem a 0))	(a i)	(i len)
6: (len <=<= 1)	(i len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen)
10: (altlen += 1)	(altlen i)	(altlen i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(ebx edi esi)	(ebx edi esi)
13: (eax <- 1)	(ebx edi esi i)	(ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a)
17: (bnds_2 <- (mem a 0))	(a x_1)	(bnds_2 x_1)
18: (bnds_2 <=<= 1)	(bnds_2 x_1)	(bnds_2 x_1)
19: (bnds_2 += 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(a i x_1)	(a i x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a x_1)	(a x_1)
26: (x_1 >>= 1)	(a x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1)
28: (x_1 += a)	(a x_1 y)	(x_1 y)
29: (aval <- (mem x_1 4))	(x_1 y)	(aval y)
30: (newaval <- aval)	(aval i y)	(i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a)
34: (bnds_4 <- (mem a 0))	(a x_3)	(bnds_4 x_3)
35: (bnds_4 <=<= 1)	(bnds_4 x_3)	(bnds_4 x_3)
36: (bnds_4 += 1)	(a bnds_4 i x_3)	(a bnds_4 i x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i x_3)	(a i x_3)
38: :oka_7	(a i x_3)	(a i x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a x_3)	(a x_3)
43: (x_3 >>= 1)	(a newaval x_3)	(a newaval x_3)
44: (x_3 *= 4)	(a newaval x_3)	(a newaval x_3)
45: (x_3 += a)	(a i newaval x_3)	(i newaval x_3)
46: ((mem x_3 4) <- newaval)	(i newaval x_3)	(i)
47: (_set <- 1)	(a i)	(a i)
48: (nexti <- i)	(a i)	(a nexti)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(nexti y)
51: (edx <- nexti)	(nexti y)	(y)

	in	out
1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax)
4: (y <- eax)	(a eax)	(a i)
5: (len <- (mem a 0))	(a i)	(i len)
6: (len <=<= 1)	(i len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen)	(altlen i)
10: (altlen += 1)	(altlen i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(ebx edi esi)	(ebx edi esi i)
13: (eax <- 1)	(ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a x_1)
17: (bnds_2 <- (mem a 0))	(a x_1)	(bnds_2 x_1)
18: (bnds_2 <=<= 1)	(bnds_2 x_1)	(a bnds_2 i x_1)
19: (bnds_2 += 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(a i x_1)	(a i x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a x_1)	(a x_1)
26: (x_1 >>= 1)	(a x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1)	(a x_1 y)
28: (x_1 += a)	(a x_1 y)	(x_1 y)
29: (aval <- (mem x_1 4))	(x_1 y)	(aval i y)
30: (newaval <- aval)	(aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a x_3)
34: (bnds_4 <- (mem a 0))	(a x_3)	(bnds_4 x_3)
35: (bnds_4 <=<= 1)	(bnds_4 x_3)	(a bnds_4 i x_3)
36: (bnds_4 += 1)	(a bnds_4 i x_3)	(a bnds_4 i x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i x_3)	(a i x_3)
38: :oka_7	(a i x_3)	(a i x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a x_3)	(a newaval x_3)
43: (x_3 >>= 1)	(a newaval x_3)	(a newaval x_3)
44: (x_3 *= 4)	(a newaval x_3)	(a i newaval x_3)
45: (x_3 += a)	(a i newaval x_3)	(i newaval x_3)
46: ((mem x_3 4) <- newaval)	(i newaval x_3)	(a i)
47: (_set <- 1)	(a i)	(a i)
48: (nexti <- i)	(a i)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(nexti y)
51: (edx <- nexti)	(nexti y)	(ecx y)

	in	out
1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax)
4: (y <- eax)	(a eax i)	(a i)
5: (len <- (mem a 0))	(a i)	(i len)
6: (len <=<= 1)	(i len)	(i len)
7: (len += 1)	(i len)	(i len)
8: (altlen <- i < len)	(i len)	(altlen)
9: (altlen += altlen)	(altlen i)	(altlen i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(ebx edi esi i)	(ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a x_1)
17: (bnds_2 <- (mem a 0))	(a x_1)	(bnds_2 x_1)
18: (bnds_2 <=<= 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
19: (bnds_2 += 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(a i x_1)	(a i x_1)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1)	(a i x_1)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a x_1)	(a x_1)
26: (x_1 >>= 1)	(a x_1)	(a x_1)
27: (x_1 *= 4)	(a x_1 y)	(a x_1 y)
28: (x_1 += a)	(a x_1 y)	(x_1 y)
29: (aval <- (mem x_1 4))	(i x_1 y)	(aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a x_3)
34: (bnds_4 <- (mem a 0))	(a x_3)	(bnds_4 x_3)
35: (bnds_4 <=<= 1)	(a bnds_4 i x_3)	(a bnds_4 i x_3)
36: (bnds_4 += 1)	(a bnds_4 i x_3)	(a bnds_4 i x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i x_3)	(a i x_3)
38: :oka_7	(a i x_3)	(a i x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i x_3)	(a i x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a newaval x_3)	(a newaval x_3)
43: (x_3 >>= 1)	(a newaval x_3)	(a newaval x_3)
44: (x_3 *= 4)	(a i newaval x_3)	(a i newaval x_3)
45: (x_3 += a)	(a i newaval x_3)	(i newaval x_3)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3)	(a i)
47: (_set <- 1)	(a i)	(a i)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(altlen i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
(a i)
(a x_1)
(a x_1)
(a x_1 y)
(a x_1 y)
(a x_1 y)
(i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i x_3)
(a i x_3)
(a i)
(a i)
(a newaval x_3)
(a newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(altlen i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
()
(a x_1)
(a x_1 y)
(a x_1 y)
(i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i x_3)
(a i newaval x_3)
(a i)
()
(a newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
(a i)
(a x_1)
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i x_3)
(a i newaval x_3)
(a i)
(a i)
(a newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(altlen i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
()
(a x_1)
(a x_1 y)
(a x_1 y)
(i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i x_3)
(a i newaval x_3)
(a i)
()
(a newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
(a i)
(a x_1)
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
()
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
()
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
(a i)
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
()
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
()
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1)
(a i)
(a i)
(a x_1 y)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1 y)
(a i)
()
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
()
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1 y)
(a i)
(a i)
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1)
(a i x_1 y)
(a i)
()
(a x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
()
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

	in	out
1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax i)
4: (y <- eax)	(a eax i)	(a i)
5: (len <- (mem a 0))	(a i)	(i len)
6: (len <<= 1)	(i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
8: (altlen <- i < len)	(a ebx edi esi i len)	(a altlen ebx edi esi i)
9: (altlen += altlen)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(a ebx edi esi i)	(a ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a i x_1)
17: (bnds_2 <- (mem a 0))	(a i x_1)	(a bnds_2 i x_1)
18: (bnds_2 <<= 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
19: (bnds_2 += 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1)	(a i x_1)
21: :oka_4	(a i x_1)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a i x_3)
34: (bnds_4 <- (mem a 0))	(a i x_3)	(a bnds_4 i x_3)
35: (bnds_4 <<= 1)	(a bnds_4 i x_3)	(a bnds_4 i x_3)
36: (bnds_4 += 1)	(a bnds_4 i x_3)	(a bnds_4 i newaval x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3)	(a i newaval x_3)
38: :oka_7	(a i newaval x_3)	(a i newaval x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3)	(a i newaval x_3)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3)	(a i newaval x_3)
43: (x_3 >>= 1)	(a i newaval x_3)	(a i newaval x_3)
44: (x_3 *= 4)	(a i newaval x_3)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(ecx nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx edx y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
()
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ecx edx)
(a eax edx)
(a eax i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax edx)
(a eax i)
(a i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a bnds_2 i x_1)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i)
(a i x_3)
(a bnds_4 i x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i newaval x_3)
(a i)
(a i)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```


1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax edx)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <=< 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
8: (altlen <- i < len)	(a ebx edi esi i len)	(a altlen ebx edi esi i)
9: (altlen += altlen)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(a ebx edi esi i)	(a ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a i x_1)
17: (bnds_2 <- (mem a 0))	(a i x_1)	(a bnds_2 i x_1)
18: (bnds_2 <=< 1)	(a bnds_2 i x_1)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i)	(a i newaval x_3)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3)	(a bnds_4 i newaval x_3)
35: (bnds_4 <=< 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3)	(a i newaval x_3)
38: :oka_7	(a i newaval x_3)	(a i newaval x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(ecx nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx edx y)

	in	out
1: :inc	(eax ecx edx)	(eax ecx edx)
2: (a <- ecx)	(eax ecx edx)	(a eax edx)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <<= 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
8: (altlen <- i < len)	(a ebx edi esi i len)	(a altlen ebx edi esi i)
9: (altlen += altlen)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(a ebx edi esi i)	(a ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a i x_1)
17: (bnds_2 <- (mem a 0))	(a i x_1)	(a bnds_2 i x_1)
18: (bnds_2 <<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i)
33: (x_3 <- i)	(a i newaval)	(a i newaval x_3)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3)	(a bnds_4 i newaval x_3)
35: (bnds_4 <<= 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3)	(a i newaval x_3)
38: :oka_7	(a i newaval x_3)	(a i newaval x_3)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(ecx nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx edx y)


```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ecx edx)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ecx edx)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

	in	out
1: :inc	(eax ecx edx)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <<= 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
8: (altlen <- i < len)	(a ebx edi esi i len)	(a altlen ebx edi esi i)
9: (altlen += altlen)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(a ebx edi esi i)	(a ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i)
16: (x_1 <- i)	(a i)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i newaval)
33: (x_3 <- i)	(a i newaval)	(a i newaval x_3)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3)	(a bnds_4 i newaval x_3)
35: (bnds_4 <<= 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(ecx nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx edx y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

	in	out
1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <=<= 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
8: (altlen <- i < len)	(a ebx edi esi i len)	(a altlen ebx edi esi i)
9: (altlen += altlen)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
10: (altlen += 1)	(a altlen ebx edi esi i)	(a altlen ebx edi esi i)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i)	(a ebx edi esi i)
12: :L_2	(a ebx edi esi i)	(a ebx edi esi i)
13: (eax <- 1)	(a ebx edi esi i)	(a ebx edi esi i)
14: (return)	(a ebx edi esi i)	(a i)
15: :L_1	(a i)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <=<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval)
32: (newaval -= 1)	(a i newaval)	(a i newaval)
33: (x_3 <- i)	(a i newaval)	(a i newaval x_3)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3)	(a bnds_4 i newaval x_3)
35: (bnds_4 <=<= 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3)	(a bnds_4 i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a nexti y)
49: (nexti += 2)	(a nexti y)	(a nexti y)
50: (ecx <- a)	(a nexti y)	(ecx nexti y)
51: (edx <- nexti)	(ecx nexti y)	(ecx edx y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i)
(a ebx edi esi i)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i y)
(a ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i)
(a ebx edi esi i)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ecx edx y)

```

1: <code>:inc</code>	<code>(eax ebx ecx edi edx esi)</code>	<code>(eax ebx ecx edi edx esi)</code>
2: <code>(a <- ecx)</code>	<code>(eax ebx ecx edi edx esi)</code>	<code>(a eax ebx edi edx esi)</code>
3: <code>(i <- edx)</code>	<code>(a eax ebx edi edx esi)</code>	<code>(a eax ebx edi esi i)</code>
4: <code>(y <- eax)</code>	<code>(a eax ebx edi esi i)</code>	<code>(a ebx edi esi i)</code>
5: <code>(len <- (mem a 0))</code>	<code>(a ebx edi esi i)</code>	<code>(a ebx edi esi i len)</code>
6: <code>(len <<= 1)</code>	<code>(a ebx edi esi i len)</code>	<code>(a ebx edi esi i len)</code>
7: <code>(len += 1)</code>	<code>(a ebx edi esi i len)</code>	<code>(a ebx edi esi i len)</code>
8: <code>(altlen <- i < len)</code>	<code>(a ebx edi esi i len)</code>	<code>(a altlen ebx edi esi i)</code>
9: <code>(altlen += altlen)</code>	<code>(a altlen ebx edi esi i)</code>	<code>(a altlen ebx edi esi i)</code>
10: <code>(altlen += 1)</code>	<code>(a altlen ebx edi esi i)</code>	<code>(a altlen ebx edi esi i y)</code>
11: <code>(cjump altlen = 1 :L_2 :L_1)</code>	<code>(a altlen ebx edi esi i y)</code>	<code>(a ebx edi esi i y)</code>
12: <code>:L_2</code>	<code>(a ebx edi esi i)</code>	<code>(a ebx edi esi i)</code>
13: <code>(eax <- 1)</code>	<code>(a ebx edi esi i)</code>	<code>(a ebx edi esi i y)</code>
14: <code>(return)</code>	<code>(a ebx edi esi i y)</code>	<code>(a i y)</code>
15: <code>:L_1</code>	<code>(a i y)</code>	<code>(a i y)</code>
16: <code>(x_1 <- i)</code>	<code>(a i y)</code>	<code>(a i x_1 y)</code>
17: <code>(bnds_2 <- (mem a 0))</code>	<code>(a i x_1 y)</code>	<code>(a bnds_2 i x_1 y)</code>
18: <code>(bnds_2 <<= 1)</code>	<code>(a bnds_2 i x_1 y)</code>	<code>(a bnds_2 i x_1 y)</code>
19: <code>(bnds_2 += 1)</code>	<code>(a bnds_2 i x_1 y)</code>	<code>(a bnds_2 i x_1 y)</code>
20: <code>(cjump x_1 < bnds_2 :oka_4 :die_3)</code>	<code>(a bnds_2 i x_1 y)</code>	<code>(a i x_1 y)</code>
21: <code>:oka_4</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
22: <code>(cjump 0 <= x_1 :okb_5 :die_3)</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
23: <code>:die_3</code>	<code>(a i)</code>	<code>(a i)</code>
24: <code>(eax <- (array-error a i))</code>	<code>(a i)</code>	<code>()</code>
25: <code>:okb_5</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
26: <code>(x_1 >>= 1)</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
27: <code>(x_1 *= 4)</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
28: <code>(x_1 += a)</code>	<code>(a i x_1 y)</code>	<code>(a i x_1 y)</code>
29: <code>(aval <- (mem x_1 4))</code>	<code>(a i x_1 y)</code>	<code>(a aval i y)</code>
30: <code>(newaval <- aval)</code>	<code>(a aval i y)</code>	<code>(a i newaval y)</code>
31: <code>(newaval += y)</code>	<code>(a i newaval y)</code>	<code>(a i newaval)</code>
32: <code>(newaval -= 1)</code>	<code>(a i newaval)</code>	<code>(a i newaval)</code>
33: <code>(x_3 <- i)</code>	<code>(a i newaval)</code>	<code>(a i newaval x_3)</code>
34: <code>(bnds_4 <- (mem a 0))</code>	<code>(a i newaval x_3)</code>	<code>(a bnds_4 i newaval x_3 y)</code>
35: <code>(bnds_4 <<= 1)</code>	<code>(a bnds_4 i newaval x_3 y)</code>	<code>(a bnds_4 i newaval x_3 y)</code>
36: <code>(bnds_4 += 1)</code>	<code>(a bnds_4 i newaval x_3 y)</code>	<code>(a bnds_4 i newaval x_3 y)</code>
37: <code>(cjump x_3 < bnds_4 :oka_7 :die_6)</code>	<code>(a bnds_4 i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
38: <code>:oka_7</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
39: <code>(cjump 0 <= x_3 :okb_8 :die_6)</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
40: <code>:die_6</code>	<code>(a i)</code>	<code>(a i)</code>
41: <code>(eax <- (array-error a i))</code>	<code>(a i)</code>	<code>()</code>
42: <code>:okb_8</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
43: <code>(x_3 >>= 1)</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
44: <code>(x_3 *= 4)</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
45: <code>(x_3 += a)</code>	<code>(a i newaval x_3 y)</code>	<code>(a i newaval x_3 y)</code>
46: <code>((mem x_3 4) <- newaval)</code>	<code>(a i newaval x_3 y)</code>	<code>(a i y)</code>
47: <code>(_set <- 1)</code>	<code>(a i y)</code>	<code>(a i y)</code>
48: <code>(nexti <- i)</code>	<code>(a i y)</code>	<code>(a nexti y)</code>
49: <code>(nexti += 2)</code>	<code>(a nexti y)</code>	<code>(a nexti y)</code>
50: <code>(ecx <- a)</code>	<code>(a nexti y)</code>	<code>(ecx nexti y)</code>
51: <code>(edx <- nexti)</code>	<code>(ecx nexti y)</code>	<code>(ebx ecx edi edx esi y)</code>

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(a nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a i y)
(a nexti y)
(a nexti y)
(ecx nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a nexti y)
(a nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval y)
(a i newaval)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a ebx edi esi i len)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval)
(a i newaval y)
(a i newaval)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```


1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <=<= 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <=<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval y)
32: (newaval -= 1)	(a i newaval y)	(a i newaval y)
33: (x_3 <- i)	(a i newaval y)	(a i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: (bnds_4 <=<= 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len)
6: (len <<= 1)	(a ebx edi esi i len)	(a ebx edi esi i len)
7: (len += 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval y)
32: (newaval -= 1)	(a i newaval y)	(a i newaval y)
33: (x_3 <- i)	(a i newaval y)	(a i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: (bnds_4 <<= 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a i y)
47: (_set <- 1)	(a i y)	(a i y)
48: (nexti <- i)	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i)
(a ebx edi esi i len)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```


1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i)	(a ebx edi esi i len y)
6: (len <<= 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: (len += 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <<= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval y)
32: (newaval -= 1)	(a i newaval y)	(a i newaval y)
33: (x_3 <- i)	(a i newaval y)	(a i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: (bnds_4 <<= 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a i newaval x_3 y)	(a ebx edi esi i y)
47: (_set <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
48: (nexti <- i)	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i)
5: (len <- (mem a 0))	(a ebx edi esi i y)	(a ebx edi esi i len y)
6: (len <=< 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: (len += 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <=< 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >>= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval y)
32: (newaval -= 1)	(a i newaval y)	(a i newaval y)
33: (x_3 <- i)	(a i newaval y)	(a i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: (bnds_4 <=< 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: :oka_7	(a i newaval x_3 y)	(a i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a i newaval x_3 y)	(a i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a i newaval x_3 y)	(a i newaval x_3 y)
43: (x_3 >>= 1)	(a i newaval x_3 y)	(a i newaval x_3 y)
44: (x_3 *= 4)	(a i newaval x_3 y)	(a i newaval x_3 y)
45: (x_3 += a)	(a i newaval x_3 y)	(a i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i y)
47: (_set <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
48: (nexti <- i)	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

	in	out
1: <code>:inc</code>	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: <code>(a <- ecx)</code>	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: <code>(i <- edx)</code>	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: <code>(y <- eax)</code>	(a eax ebx edi esi i)	(a ebx edi esi i y)
5: <code>(len <- (mem a 0))</code>	(a ebx edi esi i y)	(a ebx edi esi i len y)
6: <code>(len <=< 1)</code>	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: <code>(len += 1)</code>	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: <code>(altlen <- i < len)</code>	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: <code>(altlen += altlen)</code>	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: <code>(altlen += 1)</code>	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: <code>(cjump altlen = 1 :L_2 :L_1)</code>	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: <code>:L_2</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
13: <code>(eax <- 1)</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
14: <code>(return)</code>	(a ebx edi esi i y)	(a i y)
15: <code>:L_1</code>	(a i y)	(a i y)
16: <code>(x_1 <- i)</code>	(a i y)	(a i x_1 y)
17: <code>(bnds_2 <- (mem a 0))</code>	(a i x_1 y)	(a bnds_2 i x_1 y)
18: <code>(bnds_2 <=< 1)</code>	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: <code>(bnds_2 += 1)</code>	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: <code>(cjump x_1 < bnds_2 :oka_4 :die_3)</code>	(a bnds_2 i x_1 y)	(a i x_1 y)
21: <code>:oka_4</code>	(a i x_1 y)	(a i x_1 y)
22: <code>(cjump 0 <= x_1 :okb_5 :die_3)</code>	(a i x_1 y)	(a i x_1 y)
23: <code>:die_3</code>	(a i)	(a i)
24: <code>(eax <- (array-error a i))</code>	(a i)	()
25: <code>:okb_5</code>	(a i x_1 y)	(a i x_1 y)
26: <code>(x_1 >>= 1)</code>	(a i x_1 y)	(a i x_1 y)
27: <code>(x_1 *= 4)</code>	(a i x_1 y)	(a i x_1 y)
28: <code>(x_1 += a)</code>	(a i x_1 y)	(a i x_1 y)
29: <code>(aval <- (mem x_1 4))</code>	(a i x_1 y)	(a aval i y)
30: <code>(newaval <- aval)</code>	(a aval i y)	(a i newaval y)
31: <code>(newaval += y)</code>	(a i newaval y)	(a i newaval y)
32: <code>(newaval -= 1)</code>	(a i newaval y)	(a i newaval y)
33: <code>(x_3 <- i)</code>	(a i newaval y)	(a i newaval x_3 y)
34: <code>(bnds_4 <- (mem a 0))</code>	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: <code>(bnds_4 <=< 1)</code>	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: <code>(bnds_4 += 1)</code>	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: <code>(cjump x_3 < bnds_4 :oka_7 :die_6)</code>	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: <code>:oka_7</code>	(a i newaval x_3 y)	(a i newaval x_3 y)
39: <code>(cjump 0 <= x_3 :okb_8 :die_6)</code>	(a i newaval x_3 y)	(a i newaval x_3 y)
40: <code>:die_6</code>	(a i)	(a i)
41: <code>(eax <- (array-error a i))</code>	(a i)	()
42: <code>:okb_8</code>	(a i newaval x_3 y)	(a i newaval x_3 y)
43: <code>(x_3 >>= 1)</code>	(a i newaval x_3 y)	(a i newaval x_3 y)
44: <code>(x_3 *= 4)</code>	(a i newaval x_3 y)	(a i newaval x_3 y)
45: <code>(x_3 += a)</code>	(a i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
46: <code>((mem x_3 4) <- newaval)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i y)
47: <code>(_set <- 1)</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
48: <code>(nexti <- i)</code>	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: <code>(nexti += 2)</code>	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: <code>(ecx <- a)</code>	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: <code>(edx <- nexti)</code>	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
()
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
()
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
()
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
()
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

1: <code>:inc</code>	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: <code>(a <- ecx)</code>	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: <code>(i <- edx)</code>	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: <code>(y <- eax)</code>	(a eax ebx edi esi i)	(a ebx edi esi i y)
5: <code>(len <- (mem a 0))</code>	(a ebx edi esi i y)	(a ebx edi esi i len y)
6: <code>(len <=< 1)</code>	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: <code>(len += 1)</code>	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: <code>(altlen <- i < len)</code>	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: <code>(altlen += altlen)</code>	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: <code>(altlen += 1)</code>	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: <code>(cjump altlen = 1 :L_2 :L_1)</code>	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: <code>:L_2</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
13: <code>(eax <- 1)</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
14: <code>(return)</code>	(a ebx edi esi i y)	(a i y)
15: <code>:L_1</code>	(a i y)	(a i y)
16: <code>(x_1 <- i)</code>	(a i y)	(a i x_1 y)
17: <code>(bnds_2 <- (mem a 0))</code>	(a i x_1 y)	(a bnds_2 i x_1 y)
18: <code>(bnds_2 <=< 1)</code>	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: <code>(bnds_2 += 1)</code>	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: <code>(cjump x_1 < bnds_2 :oka_4 :die_3)</code>	(a bnds_2 i x_1 y)	(a i x_1 y)
21: <code>:oka_4</code>	(a i x_1 y)	(a i x_1 y)
22: <code>(cjump 0 <= x_1 :okb_5 :die_3)</code>	(a i x_1 y)	(a i x_1 y)
23: <code>:die_3</code>	(a i)	(a i)
24: <code>(eax <- (array-error a i))</code>	(a i)	()
25: <code>:okb_5</code>	(a i x_1 y)	(a i x_1 y)
26: <code>(x_1 >>= 1)</code>	(a i x_1 y)	(a i x_1 y)
27: <code>(x_1 *= 4)</code>	(a i x_1 y)	(a i x_1 y)
28: <code>(x_1 += a)</code>	(a i x_1 y)	(a i x_1 y)
29: <code>(aval <- (mem x_1 4))</code>	(a i x_1 y)	(a aval i y)
30: <code>(newaval <- aval)</code>	(a aval i y)	(a i newaval y)
31: <code>(newaval += y)</code>	(a i newaval y)	(a i newaval y)
32: <code>(newaval -= 1)</code>	(a i newaval y)	(a i newaval y)
33: <code>(x_3 <- i)</code>	(a i newaval y)	(a i newaval x_3 y)
34: <code>(bnds_4 <- (mem a 0))</code>	(a i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
35: <code>(bnds_4 <=< 1)</code>	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
36: <code>(bnds_4 += 1)</code>	(a bnds_4 i newaval x_3 y)	(a bnds_4 i newaval x_3 y)
37: <code>(cjump x_3 < bnds_4 :oka_7 :die_6)</code>	(a bnds_4 i newaval x_3 y)	(a i newaval x_3 y)
38: <code>:oka_7</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
39: <code>(cjump 0 <= x_3 :okb_8 :die_6)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
40: <code>:die_6</code>	(a i)	(a i)
41: <code>(eax <- (array-error a i))</code>	(a i)	()
42: <code>:okb_8</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
43: <code>(x_3 >>= 1)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
44: <code>(x_3 *= 4)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
45: <code>(x_3 += a)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
46: <code>((mem x_3 4) <- newaval)</code>	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i y)
47: <code>(_set <- 1)</code>	(a ebx edi esi i y)	(a ebx edi esi i y)
48: <code>(nexti <- i)</code>	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: <code>(nexti += 2)</code>	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: <code>(ecx <- a)</code>	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: <code>(edx <- nexti)</code>	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```


	in	out
1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i y)
5: (len <- (mem a 0))	(a ebx edi esi i y)	(a ebx edi esi i len y)
6: (len <= 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: (len += 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <= 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 i x_1 y)	(a i x_1 y)
21: :oka_4	(a i x_1 y)	(a i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a i x_1 y)	(a i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a i x_1 y)	(a i x_1 y)
26: (x_1 >= 1)	(a i x_1 y)	(a i x_1 y)
27: (x_1 *= 4)	(a i x_1 y)	(a i x_1 y)
28: (x_1 += a)	(a i x_1 y)	(a i x_1 y)
29: (aval <- (mem x_1 4))	(a i x_1 y)	(a aval i y)
30: (newaval <- aval)	(a aval i y)	(a i newaval y)
31: (newaval += y)	(a i newaval y)	(a i newaval y)
32: (newaval -= 1)	(a i newaval y)	(a i newaval y)
33: (x_3 <- i)	(a i newaval y)	(a i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
35: (bnds_4 <= 1)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
38: :oka_7	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
43: (x_3 >= 1)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
44: (x_3 *= 4)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
45: (x_3 += a)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i y)
47: (_set <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
48: (nexti <- i)	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=< 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=< 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=< 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=< 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=< 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=< 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <=<= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <=<= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >>= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <=<= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >>= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```

```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a i x_1 y)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```



```

1: :inc
2: (a <- ecx)
3: (i <- edx)
4: (y <- eax)
5: (len <- (mem a 0))
6: (len <= 1)
7: (len += 1)
8: (altlen <- i < len)
9: (altlen += altlen)
10: (altlen += 1)
11: (cjump altlen = 1 :L_2 :L_1)
12: :L_2
13: (eax <- 1)
14: (return)
15: :L_1
16: (x_1 <- i)
17: (bnds_2 <- (mem a 0))
18: (bnds_2 <= 1)
19: (bnds_2 += 1)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)
21: :oka_4
22: (cjump 0 <= x_1 :okb_5 :die_3)
23: :die_3
24: (eax <- (array-error a i))
25: :okb_5
26: (x_1 >= 1)
27: (x_1 *= 4)
28: (x_1 += a)
29: (aval <- (mem x_1 4))
30: (newaval <- aval)
31: (newaval += y)
32: (newaval -= 1)
33: (x_3 <- i)
34: (bnds_4 <- (mem a 0))
35: (bnds_4 <= 1)
36: (bnds_4 += 1)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)
38: :oka_7
39: (cjump 0 <= x_3 :okb_8 :die_6)
40: :die_6
41: (eax <- (array-error a i))
42: :okb_8
43: (x_3 >= 1)
44: (x_3 *= 4)
45: (x_3 += a)
46: ((mem x_3 4) <- newaval)
47: (_set <- 1)
48: (nexti <- i)
49: (nexti += 2)
50: (ecx <- a)
51: (edx <- nexti)

```

in

```

(eax ebx ecx edi edx esi)
(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
(a i)
(a i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
(a i)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi esi nexti y)

```

out

```

(eax ebx ecx edi edx esi)
(a eax ebx edi edx esi)
(a eax ebx edi esi i)
(a ebx edi esi i y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a ebx edi esi i len y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a altlen ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a i y)
(a i y)
(a i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a bnds_2 i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i x_1 y)
(a i)
()
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a ebx edi esi i x_1 y)
(a aval ebx edi esi i y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval y)
(a ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a bnds_4 ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a i)
()
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i newaval x_3 y)
(a ebx edi esi i y)
(a ebx edi esi i y)
(a ebx edi esi nexti y)
(a ebx edi esi nexti y)
(ebx ecx edi esi nexti y)
(ebx ecx edi edx esi y)

```


	in	out
1: :inc	(eax ebx ecx edi edx esi)	(eax ebx ecx edi edx esi)
2: (a <- ecx)	(eax ebx ecx edi edx esi)	(a eax ebx edi edx esi)
3: (i <- edx)	(a eax ebx edi edx esi)	(a eax ebx edi esi i)
4: (y <- eax)	(a eax ebx edi esi i)	(a ebx edi esi i y)
5: (len <- (mem a 0))	(a ebx edi esi i y)	(a ebx edi esi i len y)
6: (len <=< 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
7: (len += 1)	(a ebx edi esi i len y)	(a ebx edi esi i len y)
8: (altlen <- i < len)	(a ebx edi esi i len y)	(a altlen ebx edi esi i y)
9: (altlen += altlen)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
10: (altlen += 1)	(a altlen ebx edi esi i y)	(a altlen ebx edi esi i y)
11: (cjump altlen = 1 :L_2 :L_1)	(a altlen ebx edi esi i y)	(a ebx edi esi i y)
12: :L_2	(a ebx edi esi i y)	(a ebx edi esi i y)
13: (eax <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
14: (return)	(a ebx edi esi i y)	(a i y)
15: :L_1	(a i y)	(a i y)
16: (x_1 <- i)	(a i y)	(a i x_1 y)
17: (bnds_2 <- (mem a 0))	(a i x_1 y)	(a bnds_2 i x_1 y)
18: (bnds_2 <=< 1)	(a bnds_2 i x_1 y)	(a bnds_2 i x_1 y)
19: (bnds_2 += 1)	(a bnds_2 i x_1 y)	(a bnds_2 ebx edi esi i x_1 y)
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	(a bnds_2 ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
21: :oka_4	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
22: (cjump 0 <= x_1 :okb_5 :die_3)	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
23: :die_3	(a i)	(a i)
24: (eax <- (array-error a i))	(a i)	()
25: :okb_5	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
26: (x_1 >>= 1)	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
27: (x_1 *= 4)	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
28: (x_1 += a)	(a ebx edi esi i x_1 y)	(a ebx edi esi i x_1 y)
29: (aval <- (mem x_1 4))	(a ebx edi esi i x_1 y)	(a aval ebx edi esi i y)
30: (newaval <- aval)	(a aval ebx edi esi i y)	(a ebx edi esi i newaval y)
31: (newaval += y)	(a ebx edi esi i newaval y)	(a ebx edi esi i newaval y)
32: (newaval -= 1)	(a ebx edi esi i newaval y)	(a ebx edi esi i newaval y)
33: (x_3 <- i)	(a ebx edi esi i newaval y)	(a ebx edi esi i newaval x_3 y)
34: (bnds_4 <- (mem a 0))	(a ebx edi esi i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
35: (bnds_4 <=< 1)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
36: (bnds_4 += 1)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a bnds_4 ebx edi esi i newaval x_3 y)
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	(a bnds_4 ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
38: :oka_7	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
39: (cjump 0 <= x_3 :okb_8 :die_6)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
40: :die_6	(a i)	(a i)
41: (eax <- (array-error a i))	(a i)	()
42: :okb_8	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
43: (x_3 >>= 1)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
44: (x_3 *= 4)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
45: (x_3 += a)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i newaval x_3 y)
46: ((mem x_3 4) <- newaval)	(a ebx edi esi i newaval x_3 y)	(a ebx edi esi i y)
47: (_set <- 1)	(a ebx edi esi i y)	(a ebx edi esi i y)
48: (nexti <- i)	(a ebx edi esi i y)	(a ebx edi esi nexti y)
49: (nexti += 2)	(a ebx edi esi nexti y)	(a ebx edi esi nexti y)
50: (ecx <- a)	(a ebx edi esi nexti y)	(ebx ecx edi esi nexti y)
51: (edx <- nexti)	(ebx ecx edi esi nexti y)	(ebx ecx edi edx esi y)

Line 26, `_set` is not in the out set, so that's dead code;
we can drop it

```

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <<= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <<= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <<= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 += a)
((mem x_3 4) <- newaval)
(_set <- 1)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <<= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <<= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <<= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 += a)
((mem x_3 4) <- newaval)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

```

To do more with this program, we need a more sophisticated analysis.

For each statement, collect a set of expressions that have been computed and the variables they belong to.

This is something similar to **value numbering** but adapted to our setting; typically this is done with three address code, but we have two address code

This is a forwards data flow (opposite from liveness), so the in set is computed as the intersection of the outs of the predecessors.

The out set is the in set, but with adjustments to the sets made by this instruction; memory references are ignored, since don't know when two memory addresses are the same

$\text{in, out} : \text{inst\#} \rightarrow (x \rightarrow \text{expr})$

$\text{in}(1) = \emptyset$

$\text{in}(s) = \cap \{\text{out}(s') \mid s' \in \text{prec}(s)\}$

$\text{out}(s) = (\text{match } (\text{inst } s)$

$[\text{` } ((\text{mem } ,x ,y) \leftarrow ,s) \text{ in}(s)]$

$[\text{` } (,x \leftarrow (\text{mem } ,y ,z)) \text{ in}(s) \setminus x]$

$[\text{` } (,x \leftarrow ,s) \text{ in}(s) \oplus \{x \rightarrow s\}]$

$[\text{` } (,x ,\text{op} = ,y) \text{ in}(s) \oplus \{x \rightarrow (\text{in}(s)(x) \text{ op } y)\}]$

$[\text{else } \text{in}(s)])$

where \oplus replaces a mapping and \setminus removes a mapping completely

Here are the first few iterations of the computation for our program (watch what happens with \mathbf{x}_1 specifically, lines 13–16)

```

1: :inc                                0
2: (a <- ecx)                          0
3: (i <- edx)                          0
4: (y <- eax)                          0
5: (len <- (mem a 0))                  0
6: (len <<= 1)                         0
7: (len += 1)                          0
8: (altlen <- i < len)                 0
9: (altlen += altlen)                  0
10: (altlen += 1)                      0
11: (cjump altlen = 1 :L_2 :L_1)        0
12: :L_2                                0
13: (eax <- 1)                          0
14: (return)                            0
15: :L_1                                0
16: (x_1 <- i)                          0
17: (bnds_2 <- (mem a 0))                0
18: (bnds_2 <<= 1)                       0
19: (bnds_2 += 1)                        0
20: (cjump x_1 < bnds_2 :oka_4 :die_3)  0
21: :oka_4                               0
22: (cjump 0 <= x_1 :okb_5 :die_3)      0
23: :die_3                               0
24: (eax <- (array-error a i))          0
25: :okb_5                               0
26: (x_1 >>= 1)                         0
27: (x_1 *= 4)                           0
28: (x_1 += a)                           0
29: (aval <- (mem x_1 4))                 0
30: (newaval <- aval)                    0
31: (newaval += y)                       0
32: (newaval -= 1)                       0
33: (x_3 <- i)                           0
34: (bnds_4 <- (mem a 0))                0
35: (bnds_4 <<= 1)                       0
36: (bnds_4 += 1)                        0
37: (cjump x_3 < bnds_4 :oka_7 :die_6)  0
38: :oka_7                               0
39: (cjump 0 <= x_3 :okb_8 :die_6)      0
40: :die_6                               0
41: (eax <- (array-error a i))          0
42: :okb_8                               0
43: (x_3 >>= 1)                         0
44: (x_3 *= 4)                           0
45: (x_3 += a)                           0
46: ((mem x_3 4) <- newaval)             0
47: (nexti <- i)                         0
48: (nexti += 2)                         0
49: (ecx <- a)                           0
50: (edx <- nexti)                       0
51: (eax <- y)                           0
52: (goto :inc)                          0

```

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	()	{{edx i}}
4: (y <- eax)	()	{{eax y}}
5: (len <- (mem a 0))	()	()
6: (len <<= 1)	()	()
7: (len += 1)	()	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{1 eax}}
14: (return)	()	()
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	()	()
18: (bnds_2 <<= 1)	()	()
19: (bnds_2 += 1)	()	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	()	()
32: (newaval -= 1)	()	()
33: (x_3 <- i)	()	{{i x_3}}
34: (bnds_4 <- (mem a 0))	()	()
35: (bnds_4 <<= 1)	()	()
36: (bnds_4 += 1)	()	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	()	()
49: (ecx <- a)	()	{{a ecx}}
50: (edx <- nexti)	()	{{nexti edx}}
51: (eax <- y)	()	{{y eax}}
52: (goto :inc)	()	()

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i}}
4: (y <- eax)	{{edx i}}	{{eax y}}
5: (len <- (mem a 0))	{{eax y}}	()
6: (len <<= 1)	()	()
7: (len += 1)	()	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{1 eax}}
14: (return)	{{1 eax}}	()
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	()
18: (bnds_2 <<= 1)	()	()
19: (bnds_2 += 1)	()	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	()
32: (newaval -= 1)	()	()
33: (x_3 <- i)	()	{{i x_3}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	()
35: (bnds_4 <<= 1)	()	()
36: (bnds_4 += 1)	()	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	()
49: (ecx <- a)	()	{{a ecx}}
50: (edx <- nexti)	{{a ecx}}	{{nexti edx}}
51: (eax <- y)	{{nexti edx}}	{{y eax}}
52: (goto :inc)	{{y eax}}	()

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx i}}	{{edx i} {ecx a}}
4: (y <- eax)	{{edx i}}	{{eax y} {edx i}}
5: (len <- (mem a 0))	{{eax y}}	{{eax y}}
6: (len <<= 1)	()	()
7: (len += 1)	()	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{! eax}}
14: (return)	{{! eax}}	{{! eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	()	()
19: (bnds_2 += 1)	()	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	()	()
33: (x_3 <- i)	()	{{i x_3}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	()	()
36: (bnds_4 += 1)	()	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	()	{{a ecx}}
50: (edx <- nexti)	{{a ecx}}	{{nexti edx} {a ecx}}
51: (eax <- y)	{{nexti edx}}	{{y eax} {nexti edx}}
52: (goto :inc)	{{y eax}}	{{y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {edx i}}
5: (len <- (mem a 0))	{{eax y} {edx i}}	{{eax y}}
6: (len <<= 1)	{{eax y}}	()
7: (len += 1)	()	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{! eax}}
14: (return)	{{! eax}}	{{! eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	()
19: (bnds_2 += 1)	()	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	()
33: (x_3 <- i)	()	{{i x_3}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	()
36: (bnds_4 += 1)	()	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx}}
50: (edx <- nexti)	{{a ecx}}	{{nexti edx} {a ecx}}
51: (eax <- y)	{{a ecx} {nexti edx}}	{{y eax} {nexti edx}}
52: (goto :inc)	{{nexti edx} {y eax}}	{{y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
5: (len <- (mem a 0))	{{eax y} {edx i}}	{{eax y} {edx i}}
6: (len <<= 1)	{{eax y}}	{{eax y}}
7: (len += 1)	()	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{! eax}}
14: (return)	{{! eax}}	{{! eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	{{i x_1}}
19: (bnds_2 += 1)	()	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	{{((aval + y) - 1) newaval}}
33: (x_3 <- i)	()	{{i x_3}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	{{i x_3}}
36: (bnds_4 += 1)	()	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx} {(i + 2) nexti}}
50: (edx <- nexti)	{{a ecx}}	{{nexti edx} {a ecx}}
51: (eax <- y)	{{a ecx} {nexti edx}}	{{y eax} {a ecx} {nexti edx}}
52: (goto :inc)	{{nexti edx} {y eax}}	{{nexti edx} {y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
5: (len <- (mem a 0))	{{eax y} {ecx a} {edx i}}	{{eax y} {edx i}}
6: (len <<= 1)	{{eax y} {edx i}}	{{eax y}}
7: (len += 1)	{{eax y}}	()
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{! eax}}
14: (return)	{{! eax}}	{{! eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	{{i x_1}}
19: (bnds_2 += 1)	{{i x_1}}	()
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	{{((aval + y) - 1) newaval}}
33: (x_3 <- i)	{{((aval + y) - 1) newaval}}	{{i x_3}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	{{i x_3}}
36: (bnds_4 += 1)	{{i x_3}}	()
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx} {(i + 2) nexti}}
50: (edx <- nexti)	{{(i + 2) nexti} {a ecx}}	{{nexti edx} {a ecx}}
51: (eax <- y)	{{a ecx} {nexti edx}}	{{y eax} {a ecx} {nexti edx}}
52: (goto :inc)	{{a ecx} {nexti edx} {y eax}}	{{nexti edx} {y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
5: (len <- (mem a 0))	{{eax y} {ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
6: (len <<= 1)	{{eax y} {edx i}}	{{eax y} {edx i}}
7: (len += 1)	{{eax y}}	{{eax y}}
8: (altlen <- i < len)	()	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{1 eax}}
14: (return)	{{1 eax}}	{{1 eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	{{i x_1}}
19: (bnds_2 += 1)	{{i x_1}}	{{i x_1}}
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	()	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	{{((aval + y) - 1) newaval}}
33: (x_3 <- i)	{{((aval + y) - 1) newaval}}	{{i x_3} {{(aval + y) - 1) newaval}}
34: (bnds_4 <- (mem a 0))	{{i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	{{i x_3}}
36: (bnds_4 += 1)	{{i x_3}}	{{i x_3}}
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	()	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx} {{(i + 2) nexti}}
50: (edx <- nexti)	{{(i + 2) nexti} {a ecx}}	{{nexti edx} {{(i + 2) nexti} {a ecx}}
51: (eax <- y)	{{a ecx} {nexti edx}}	{{y eax} {a ecx} {nexti edx}}
52: (goto :inc)	{{a ecx} {nexti edx} {y eax}}	{{a ecx} {nexti edx} {y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
5: (len <- (mem a 0))	{{eax y} {ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
6: (len <<= 1)	{{eax y} {ecx a} {edx i}}	{{eax y} {edx i}}
7: (len += 1)	{{eax y} {edx i}}	{{eax y}}
8: (altlen <- i < len)	{{eax y}}	()
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{1 eax}}
14: (return)	{{1 eax}}	{{1 eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	{{i x_1}}
19: (bnds_2 += 1)	{{i x_1}}	{{i x_1}}
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	{{i x_1}}	()
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	{{((aval + y) - 1) newaval}}
33: (x_3 <- i)	{{((aval + y) - 1) newaval}}	{{i x_3} {{{(aval + y) - 1} newaval}}
34: (bnds_4 <- (mem a 0))	{{((aval + y) - 1) newaval} {i x_3}}	{{i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	{{i x_3}}
36: (bnds_4 += 1)	{{i x_3}}	{{i x_3}}
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	{{i x_3}}	()
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx} {(i + 2) nexti}}
50: (edx <- nexti)	{{(i + 2) nexti} {a ecx}}	{{nexti edx} {(i + 2) nexti} {a ecx}}
51: (eax <- y)	{{(i + 2) nexti} {a ecx} {nexti edx}}	{{y eax} {a ecx} {nexti edx}}
52: (goto :inc)	{{a ecx} {nexti edx} {y eax}}	{{a ecx} {nexti edx} {y eax}}

1: :inc	()	()
2: (a <- ecx)	()	{{ecx a}}
3: (i <- edx)	{{ecx a}}	{{edx i} {ecx a}}
4: (y <- eax)	{{ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
5: (len <- (mem a 0))	{{eax y} {ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
6: (len <<= 1)	{{eax y} {ecx a} {edx i}}	{{eax y} {ecx a} {edx i}}
7: (len += 1)	{{eax y} {edx i}}	{{eax y} {edx i}}
8: (altlen <- i < len)	{{eax y}}	{{eax y}}
9: (altlen += altlen)	()	()
10: (altlen += 1)	()	()
11: (cjump altlen = 1 :L_2 :L_1)	()	()
12: :L_2	()	()
13: (eax <- 1)	()	{{! eax}}
14: (return)	{{! eax}}	{{! eax}}
15: :L_1	()	()
16: (x_1 <- i)	()	{{i x_1}}
17: (bnds_2 <- (mem a 0))	{{i x_1}}	{{i x_1}}
18: (bnds_2 <<= 1)	{{i x_1}}	{{i x_1}}
19: (bnds_2 += 1)	{{i x_1}}	{{i x_1}}
20: (cjump x_1 < bnds_2 :oka_4 :die_3)	{{i x_1}}	{{i x_1}}
21: :oka_4	()	()
22: (cjump 0 <= x_1 :okb_5 :die_3)	()	()
23: :die_3	()	()
24: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
25: :okb_5	()	()
26: (x_1 >>= 1)	()	()
27: (x_1 *= 4)	()	()
28: (x_1 += a)	()	()
29: (aval <- (mem x_1 4))	()	()
30: (newaval <- aval)	()	{{aval newaval}}
31: (newaval += y)	{{aval newaval}}	{{(aval + y) newaval}}
32: (newaval -= 1)	{{(aval + y) newaval}}	{{((aval + y) - 1) newaval}}
33: (x_3 <- i)	{{((aval + y) - 1) newaval}}	{{i x_3} {{(aval + y) - 1} newaval}}
34: (bnds_4 <- (mem a 0))	{{((aval + y) - 1) newaval} {i x_3}}	{{((aval + y) - 1) newaval} {i x_3}}
35: (bnds_4 <<= 1)	{{i x_3}}	{{i x_3}}
36: (bnds_4 += 1)	{{i x_3}}	{{i x_3}}
37: (cjump x_3 < bnds_4 :oka_7 :die_6)	{{i x_3}}	{{i x_3}}
38: :oka_7	()	()
39: (cjump 0 <= x_3 :okb_8 :die_6)	()	()
40: :die_6	()	()
41: (eax <- (array-error a i))	()	{{(array-error a i) eax}}
42: :okb_8	()	()
43: (x_3 >>= 1)	()	()
44: (x_3 *= 4)	()	()
45: (x_3 += a)	()	()
46: ((mem x_3 4) <- newaval)	()	()
47: (nexti <- i)	()	{{i nexti}}
48: (nexti += 2)	{{i nexti}}	{{(i + 2) nexti}}
49: (ecx <- a)	{{(i + 2) nexti}}	{{a ecx} {(i + 2) nexti}}
50: (edx <- nexti)	{{(i + 2) nexti} {a ecx}}	{{nexti edx} {(i + 2) nexti} {a ecx}}
51: (eax <- y)	{{(i + 2) nexti} {a ecx} {nexti edx}}	{{y eax} {(i + 2) nexti} {a ecx} {nexti edx}}
52: (goto :inc)	{{a ecx} {nexti edx} {y eax}}	{{a ecx} {nexti edx} {y eax}}

Here are the complete results for the available expressions, in tabular form

We can use this analysis to rewrite a program to use the result of an earlier computation, instead of a new computation

More precisely, if we have $(x \text{ op} = s)$ and the expression for x in the out set of that instruction is the same as the expression for some other variable y , then we can rewrite the instruction to $(x \leftarrow y)$


```

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 += a)
((mem x_3 4) <- newaval)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 <- x_1)
((mem x_3 4) <- newaval)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

```

Now we can use dead code elimination to drop the earlier computations

```

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 >>= 1)
(x_3 *= 4)
(x_3 <- x_1)
((mem x_3 4) <- newaval)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

:inc
(a <- ecx)
(i <- edx)
(y <- eax)
(len <- (mem a 0))
(len <= 1)
(len += 1)
(altlen <- i < len)
(altlen += altlen)
(altlen += 1)
(cjump altlen = 1 :L_2 :L_1)
:L_2
(eax <- 1)
(return)
:L_1
(x_1 <- i)
(bnds_2 <- (mem a 0))
(bnds_2 <= 1)
(bnds_2 += 1)
(cjump x_1 < bnds_2 :oka_4 :die_3)
:oka_4
(cjump 0 <= x_1 :okb_5 :die_3)
:die_3
(eax <- (array-error a i))
:okb_5
(x_1 >>= 1)
(x_1 *= 4)
(x_1 += a)
(aval <- (mem x_1 4))
(newaval <- aval)
(newaval += y)
(newaval -= 1)
(x_3 <- i)
(bnds_4 <- (mem a 0))
(bnds_4 <= 1)
(bnds_4 += 1)
(cjump x_3 < bnds_4 :oka_7 :die_6)
:oka_7
(cjump 0 <= x_3 :okb_8 :die_6)
:die_6
(eax <- (array-error a i))
:okb_8
(x_3 <- x_1)
((mem x_3 4) <- newaval)
(nexti <- i)
(nexti += 2)
(ecx <- a)
(edx <- nexti)
(eax <- y)
(goto :inc)

```