Josephus problem

```
01  N          IS      24                          initial population n < 256
02  M          IS      11                          every mth person is executed; m > 1
03  t          IS      $255
04  tt         GREG    0                           two additional temporary registers
05  ttt        GREG    0
06  sno        IS      $0                          Sequence number
07  cnt        IS      $1                          A counter
08  cur        IS      $2                          Current element
09  b          IS      $3                          Number of empty slots in out
10  out        IS      $4                          Output 8 characters
11  _mn        GREG    -N+1
12  _ones      GREG    #0101010101010101
13             LOC     Data_Segment
14  _buf       GREG    @
15             LOC     @+920                       maximal 2*9+3*90+4*157 = 916 bytes
16  _ctop      GREG    @+N-1                       top of circle for N bytes
17             LOC     #100
18  Main       SET     cnt,_mn                 1   Store distance to next man in circle.
19  0H         STO     _ones,_ctop,cnt     ⌈n/8⌉   Link each byte with next byte.
20             ADD     cnt,cnt,8           ⌈n/8⌉   8 links done.
21             PBNP    cnt,0B              ⌈n/8⌉
22             STB     _mn,_ctop,0             1   Last byte jumps back to the first.
23             SET     sno,0                   1   Start execution; no one excluded so far.
24             SET     cur,_mn                 1   Start with first person.
25  1H         SET     cnt,M-2             n-1     Count down m-2 links.
26  2H         LDB     tt,_ctop,cur            P   Get distance from cur to next person
27             ADD     cur,cur,tt              P     and follow the link to this person.
28             SUB     cnt,cnt,1               P
29             PBNZ    cnt,2B                  P
30             LDB     tt,_ctop,cur        n-1     cur survives the round,
31             ADD     cnt,cur,tt          n-1       but the next man cnt is executed.
32             LDB     t,_ctop,cnt         n-1     Get his neighbor.
33             ADD     tt,tt,t             n-1     Add it to previous distance for cur.
34             STB     tt,_ctop,cur        n-1     Store the new value for cur.
35             ADD     cur,cnt,t           n-1     Move to next man for next round.
36             ADD     sno,sno,1           n-1     Increase the sequence number
37             STB     sno,_ctop,cnt       n-1       and store it for the executed man.
38             CMP     t,sno,N-1           n-1     Is only one person left?
39             PBN     t,1B                n-1     If not start count down again.
40             SET     sno,N                   1   He is the last man
41             STB     sno,_ctop,cur           1     and gets the number n.
42  Output     SET     sno,_mn                 1   Start the output with first man.
43             SET     b,8                     1   8 chars in out are empty.
44             SET     cnt,0                   1   Offset in _buf to store out.
45  0H         LDB     t,_ctop,sno             n   Load next sequence number.
46  2H         DIV     t,t,10                  D   Extract
47             GET     tt,rR                   D     the digits
48             ADD     tt,tt,'0'               D     from right
49             SL      ttt,ttt,8               D     to left
50             OR      ttt,ttt,tt              D     and store them in the register ttt.
51  4H         PBNZ    t,2B                    D
52             SET     t,' '                   n   Add a space in front of each sequence number.
```

```
53              JMP     9F                      n
54  1H          AND     t,ttt,#ff               D       Move ttt in reverse order to out.
55              SR      ttt,ttt,8               D
56  9H          SL      out,out,8               D+n
57              ADD     out,out,t               D+n
58              SUB     b,b,1                   D+n
59              PBNZ    b,3F                    D+n
60              STO     out,_buf,cnt            ⌊(D+n)/8⌋ Store the characters into _buf.
61              ADD     cnt,cnt,8               ⌊(D+n)/8⌋
62              SRU     out,out,64              ⌊(D+n)/8⌋
63              SET     b,8                     ⌊(D+n)/8⌋
64  3H          PBNZ    ttt,1B                  D+n
65              ADD     sno,sno,1               n       Next sequence number.
66              PBNP    sno,0B                  n
67              SL      out,out,8               1       Add null byte to the final octabyte.
68              STO     out,_buf,cnt            1
69              LDA     t,_buf                  1       Output sequence numbers.
70              TRAP    0,Fputs,StdOut          1
71              TRAP    0,Halt,0                1       ∎
```

Analysis

The output of the programm is: `15 12 22 8 16 11 23 21 3 5 1 17 10 7 24 19 20 18 9 14 4 2 13 6`. So the last man is at position 15. The statistics at the end of the run is: `1879 instructions, 336 mems, 4350 oops; 341 good guesses, 81 bad`.

In general the program needs $(5n + \lceil n/8\rceil + \lfloor (D+n)/8\rfloor + P - 1)\mu + (27n + 3\lceil n/8\rceil + 6\lfloor (D+n)/8\rfloor + 4P + 72D + 15)\upsilon$. The value P stands for $n-1$ count downs of $m-2$ steps, that is $P = (n-1)(m-2)$. The quantity D is the number of decimal digits in the output:

$$D = \sum_{i=0}^{\lfloor \log_{10} n\rfloor - 1} 9 \cdot 10^i + (\lfloor \log_{10} n\rfloor + 1)(n + 1 - 10^{\lfloor \log_{10} n\rfloor}).$$

In the case $n = 24$, $m = 11$ the values of P and D are $23 \cdot 9 = 207$ and $9 + 2 \cdot 15 = 39$. Therefore in this case the program would need $(120 + 3 + 7 + 207 - 1)\mu = 336\mu$ and $(648 + 9 + 42 + 828 + 2808 + 15)\upsilon = 4350\upsilon$ which agrees with the measured data.