### Josephus problem

01 N IS 24  
02 M IS 11  
03 t IS $255  
04 tt GREG 0  
05 ttt GREG 0  
06 sno IS $0  
07 cnt IS $1  
08 cur IS $2  
09 b IS $3  
10 out IS $4  
11 _mn GREG -N+1  
12 _ones GREG #0101010101010101  
13 LOC Data_Segment  
14 _buf GREG @  
15 LOC @+920  
16 _ctop GREG @+N-1  
17 LOC #100  
18 Main SET cnt,_mn  
19 0H STO _ones,_ctop,cnt  
20 ADD cnt,cnt,8  
21 PBNP cnt,0B  
22 STB _mn,_ctop,0  
23 SET sno,0  
24 SET cur,_mn  
25 1H SET cnt,M-2  
26 2H LDB tt,_ctop,cur  
27 ADD cur,cur,tt  
28 SUB cnt,cur,1  
29 PENZ cnt,2B  
30 LDB tt,_ctop,cur  
31 ADD cnt,cur,tt  
32 LDB t,_ctop,cnt  
33 ADD tt,tt,t  
34 STB tt,_ctop,cur  
35 ADD cur,cnt,t  
36 ADD sno,sno,1  
37 STB sno,_ctop,cnt  
38 CMP t,sno,N-1  
39 PBNN t,1B  
40 SET sno,N  
41 STB _mn,_ctop,cnt  
42 Output SET sno,_mn  
43 SET b,8  
44 SET cnt,0  
45 0H LDB t,_ctop,sno  
46 2H DIV t,t,10  
47 GET tt,rR  
48 ADD tt,tt,'0'  
49 SL ttt,ttt,8  
50 OR ttt,ttt,tt  
51 4H PENZ t,2B  
52 SET t,' '  

---

**initial population** \( n < 256 \)  
**every** \( m \)th person is executed; \( m > 1 \)  
**two additional temporary registers**  
**Sequence number**  
**A counter**  
**Number of empty slots in** \( out \)  
**Output 8 characters**  
**maximal** \( 2 \times 9 + 3 \times 90 + 4 \times 157 = 916 \) bytes  
**top of circle** for \( N \) bytes  
**Store distance to next man in circle.**  
**Link each byte with next byte.**  
**\( \lceil n/8 \rceil \) 8 links done.**  
**Last byte jumps back to the first.**  
**Start execution; no one excluded so far.**  
**Start with first person.**  
**\( n - 1 \) Count down \( m - 2 \) links.**  
**\( n - 1 \) cur survives the round,**  
**but the next man cnt is executed.**  
**Get his neighbor.**  
**Add it to previous distance for cur.**  
**Store the new value for cur.**  
**Move to next man for next round.**  
**Increase the sequence number**  
**and store it for the executed man.**  
**Is only one person left?**  
**If not start count down again.**  
**He is the last man**  
**and gets the number \( n \).**  
**Start the output with first man.**  
**8 chars in \( out \) are empty.**  
**Offset in _buf to store out.**  
**Load next sequence number.**  
**Extract the digits**  
**from right**  
**to left**  
**and store them in the register ttt.**  
**Add a space in front of each sequence number.**
Analysis

The output of the program is: 15 12 22 8 16 11 23 21 3 5 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 \((5n + \lceil n/8 \rceil + \lfloor (D + n)/8 \rfloor + P - 1)\mu + (27n + 3\lceil n/8 \rceil + 6\lfloor (D + n)/8 \rfloor + 4P + 72D + 15)\nu\). 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:

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

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 \((648 + 9 + 42 + 828 + 2808 + 15)\nu = 4350\nu\) which agrees with the measured data.