Write, assemble and run successfully on the simulator a Mac-1 subroutine \( \text{maxm}(n,x) \) that returns in the AC the address of the integer possessing the largest absolute value (i.e., magnitude) among the \( n \) integers in the array whose starting address is \( x \). Your subroutine should be tested with the main program shown below, which defines how the parameters are passed.

```
/main program | /continued from below halt
EXTRN maxm | data 57
ans1 RES 1 | 0
ans2 RES 1 | 129
ans3 RES 1 | 8
n1 7 | -134
n2 10 | 3
n3 6 | -2
start loco 4020 | -29
swap /initialize sp | -3
loco n1 | 347
push /push address n1 | 15
loco data | -6
push /push array start address | -435
one call maxm | 13
stod ans1 | END start
insp 2
loco n2 /push address n2 |
push |
loco data |
add (4) |
push /push array start address |
two call maxm |
stod ans2 |
insp 2
loco n3 /push address n3 |
push |
loco data |
add (6) |
push /push array start address |
three call maxm |
stod ans3 |
insp 2
halt |
```

Hand in a copy of the main program symbolic assembly listing, the subroutine symbolic assembly listing, the contents of (macro) memory after “load main sub” (i.e., of main.abs) before execution of the program, and the contents of memory after execution of the program. Highlight and comment upon the final answers. Specify what values are contained in the addresses specified by ans1, ans2, and ans3.