Write, assemble and run successfully on the simulator a Mac-1 subroutine \texttt{minod(n,x)} that returns in the AC the address of the integer possessing the algebraically smallest odd value (zero and multiples of 2 are even) 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
EXTRN minod

ans1 RES 1 | data  57
ans2 RES 1 |     0
ans3 RES 1 |    129
n1     9   |    8
n2    10   |    3
n3     5   |   -29
start loco 4020 |   -2
    swap /initialize sp
    loco n1  |  -3
    push /push address n1
    loco data | 347
    push /push array start address
one    call minod |     -413
    stod ans1 |     END
    insp 2   |
    loco n2 /push address n2
    push     |
    loco data|
    add (4) |
    push /push array start address

two    call minod |
    stod ans2 |
    insp 2   |
    loco n3 /push address n3
    push     |
    loco data|
    add (7) |
    push /push array start address
three  call minod |
    stod ans3 |
    insp 2   |
    halt |

/data array continues here but
/ is shown in the above right hand column

```

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 \( \text{ans1} \), \( \text{ans2} \), and \( \text{ans3} \).