Write, assemble and run successfully on the simulator a Mac-1 subroutine `minev(n,x)` that returns in the AC the address of the integer possessing the algebraically smallest even value (i.e., the farthest left value on the real line that is a multiple of 2, including zero) among the n integers in the array whose starting address is x. If there is more than one minimum even value in the array range, return the address of the one with the greatest address value. If there are no even values in the array range, return minus one which corresponds to unsigned 65535, a clearly out of range address in the 4096 word address space. Your subroutine should be tested with the main program shown below, which defines how the parameters are passed.

```
START: loco 4020
swap /initialize sp
loco n1
push /push address n1
loco data
push /push array start address
one call minev
stod ans1
insp 2
loco n2 /push address n2
push
loco data
add (3)
push /push array start address
two call minev
stod ans2
insp 2
loco n3 /push address n3
push
loco data
add (7)
push /push array start address
three call minev
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.