Write, assemble and run successfully on the simulator a Mac-1 subroutine \texttt{minev(n, x)} that returns in the AC the address of the integer possessing the 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.

/main program  |  /continued from below halt  
\texttt{EXTRN minev}  |  data 41  
\texttt{ans1} RES 1  |  7  
\texttt{ans2} RES 1  |  129  
\texttt{ans3} RES 1  |  3  
\texttt{n1} 5  |  -133  
\texttt{n2} 9  |  0  
\texttt{n3} 8  |  -2  
\texttt{start} loco 4020  |  -29  
\texttt{swap} /initialize sp  |  -6  
\texttt{loco n1}  |  347  
\texttt{push} /push address \texttt{n1}  |  15  
\texttt{loco data}  |  -3  
\texttt{push} /push array start address  |  -435  
\texttt{one} call \texttt{minev}  |  -6  
\texttt{stod ans1}  |  END start  
\texttt{insp 2}  |  \texttt{end}  
\texttt{loco n2} /push address \texttt{n2}  |  \texttt{start}  
\texttt{push}  |  \texttt{end}  
\texttt{loco data}  |  \texttt{end}  
\texttt{add (4)}  |  \texttt{end}  
\texttt{push} /push array start address  |  \texttt{end}  
\texttt{two} call \texttt{minev}  |  \texttt{end}  
\texttt{stod ans2}  |  \texttt{end}  
\texttt{insp 2}  |  \texttt{end}  
\texttt{loco n3} /push address \texttt{n3}  |  \texttt{end}  
\texttt{push}  |  \texttt{end}  
\texttt{loco data}  |  \texttt{end}  
\texttt{add (6)}  |  \texttt{end}  
\texttt{push} /push array start address  |  \texttt{end}  
\texttt{three} call \texttt{minev}  |  \texttt{end}  
\texttt{stod ans3}  |  \texttt{end}  
\texttt{insp 2}  |  \texttt{end}  
\texttt{halt}  |  \texttt{end}  

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 \texttt{ans1}, \texttt{ans2}, and \texttt{ans3}.