Implement an interpreter/simulator for the below gven language

DATA A /\* this will be allocating one byte for A \*/

DATA B[5] /\* this will be allocating 5 bytes for B \*/

DATA D /\* this will be allocating one byte for D \*/

CONST C = 5 /\* this will make constant 5 assigned to C \*/

CONST E = 0 /\* this will make constant 0 assigned to E \*/

/\* there are 8 registers AX, BX, CX, DX, EX, FX, GX, HX \*/

/\* any arithmetic operation can be done only using registers \*/

/\* you get an instruction called MOV to move between registers or between register and memory \*/

MOV AX, C /\* now AX has value 5 \*/

MOV BX, A /\* whatever value A variable has will move to BX \*/

MOV CX, B[0] /\* whatever value B[0] has will move to CX \*/

MOV FX, E /\* moves value of E to FX \*/

MOV C, DX /\* value of DX moves to C \*/

MOV b[4], EX /\* values of EX moves to B[4] \*/

/\* arithmetic operators that we support are ADD, SUB, MUL \*/

ADD DX, AX, BX /\* AX and BX will get added and result goes to DX \*/

SUB EX, DX, CX /\* EX = DX - CX \*/

/\* logic operations IF THEN ELSE with checks GT, LT, EQ, GTEQ, LTEQ are given \*/

IF EX EQ FX THEN ADD EX, DX, CX

ELSE SUB EX, DX, CX

/\* in addition you get a JMP which can jump to the marker put in the program \*/

X:

MOV AX, C

JMP X: /\* will jump the program execution to X \*/

/\* you get two i/o operators in addition to these \*/

READ AX /\* which reads from scanf of your program and assigns to the register AX\*/

               /\* to read array of values you can use a loop construnt using JMP \*/

PRINT AX /\* you can print the values of AX using printf of C \*/

                /\* to print array again use JMP and IF to print array \*/