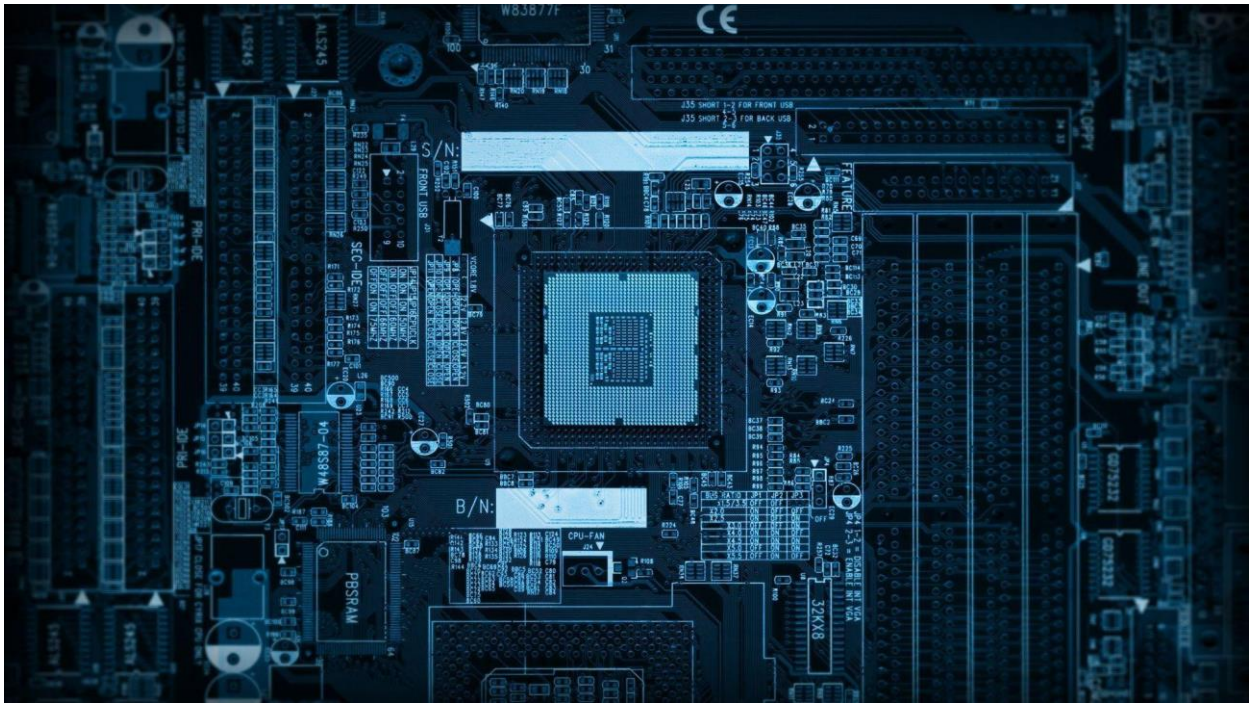


SOURCE: https://wallpaperaccess.com/electronic-circuit-board#google_vignette

19AIE101 ELEMENTS OF COMPUTING SYSTEMS-1
END SEM PROJECT REPORT

Group Members:

NAME:	ROLL NUMBER:
GOURI S VIPIN	AM.EN.U4AIE21132
VARALAKSHMI M P	AM.EN.U4AIE21165
GAYATHRY M WARIYAR	AM.EN.U4AIE21130
ANNAPOORNA A K	AM.EN.U4AIE21114
ANAGHA VIJU	AM.EN.U4AIE21112

AMRITA SCHOOL OF ENGINEERING
AMRITA VISHWA VIDYAPEETHAM
AMRITAPURI 690 525
January 2022

PSEUDO CODE:

```
// for(i = 0 ; i < 8 ; i++){
//         max = RAM[i]
//         for(j = 0 ; j < 38 ; j ++){
//         if (max == 0){skip the loop}
//         if (max - RAM[j] > 0){swap the values}


j = 30


LOOP_FROM:
      max = RAM[i]
      if (i == 8) goto END


      i = i + 1
      j = 30


LOOP_TO:


      if ((max - j) > 0) goto CHANGE


      if (max == 0) goto LOOP_FROM


      if (j == 37) goto LOOP_FROM
```

```
      j = j + 1


CHANGE:


      temp = j
      j = max
      max = temp


      goto LOOP_TO


END:


      goto END
```

HACK ASSEMBLY CODE:

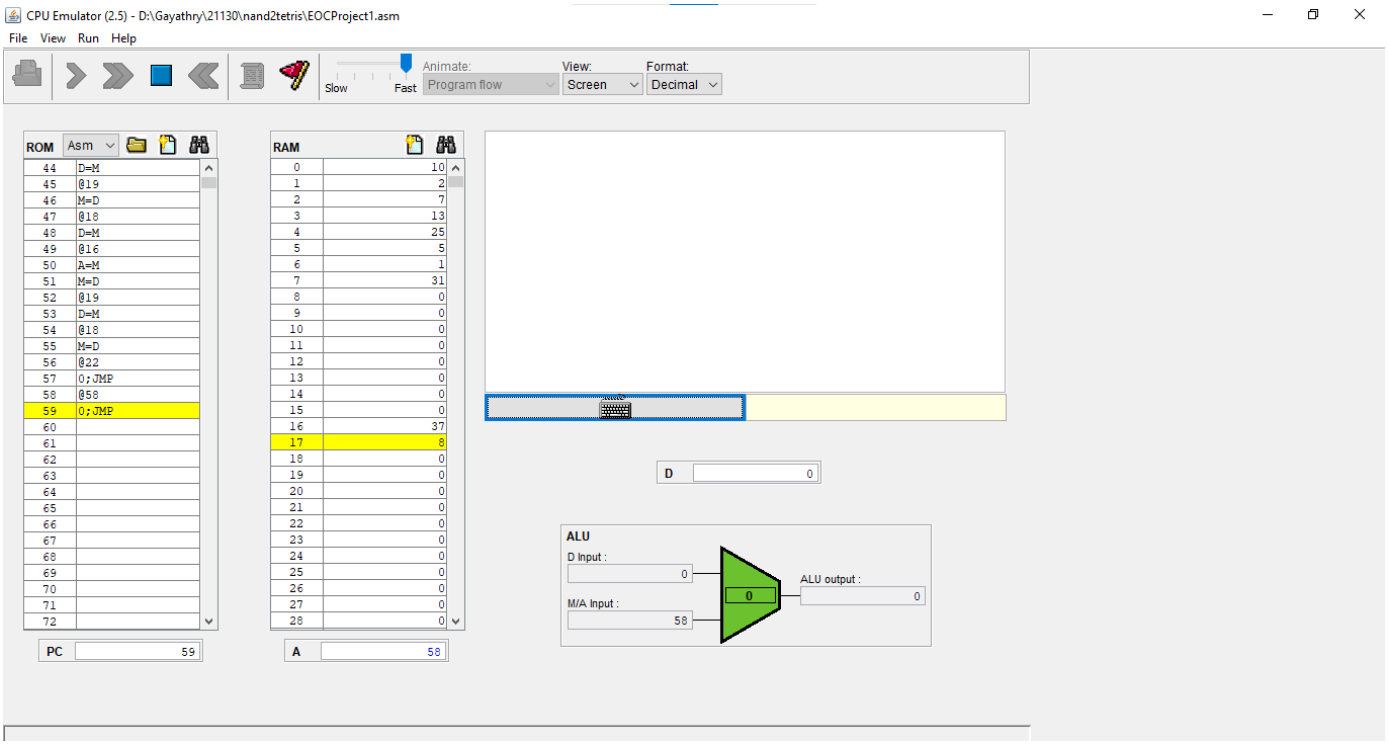
```
// asm program to sort numbers from RAM[0] to RAM[7] in descending
order and store it to RAM[30] to RAM[37]

      @30 // resetting to 30
      D=A
      @j
      M=D

(LOOP_FROM)

      @i  // variable to iterate through RAM[0] to RAM[7]
      D=M // finds the latest from address
      A=D // changing to the latest address
      D=M
      @max
      M=D // stores the number at the latest from address as the
maximum number

      @i  // loop breaking statement
      D=M
      @8
      D=D-A
      @END
      D; JEQ

      @i   // incrementing the variable for loop from
      M=M+1

      @30 // resetting to 30
      D=A
      @j
      M=D

(LOOP_TO)

      @max
      D=M
```

```
      @j
      A=M

      D=D-M // if max is greater, D will be +ve

      @CHANGE // if max is greater, numbers are interchanged
      D;JGT

     @max // if max == 0, breaks the loop and go with next number
      D=M

      @LOOP_FROM
      D;JEQ

      @j  // loop breaking statement
      D=M
      @37
      D=D-A
      @LOOP_FROM
      D;JEQ



      @j   // incrementing the variable for loop from
      M=M+1

(CHANGE)
      @j   // from RAM[j] to temp
      D=M
      A=D
      D=M
      @temp
      M=D


      @max // from max to j
      D=M
      @j
      A=M
      M=D
```

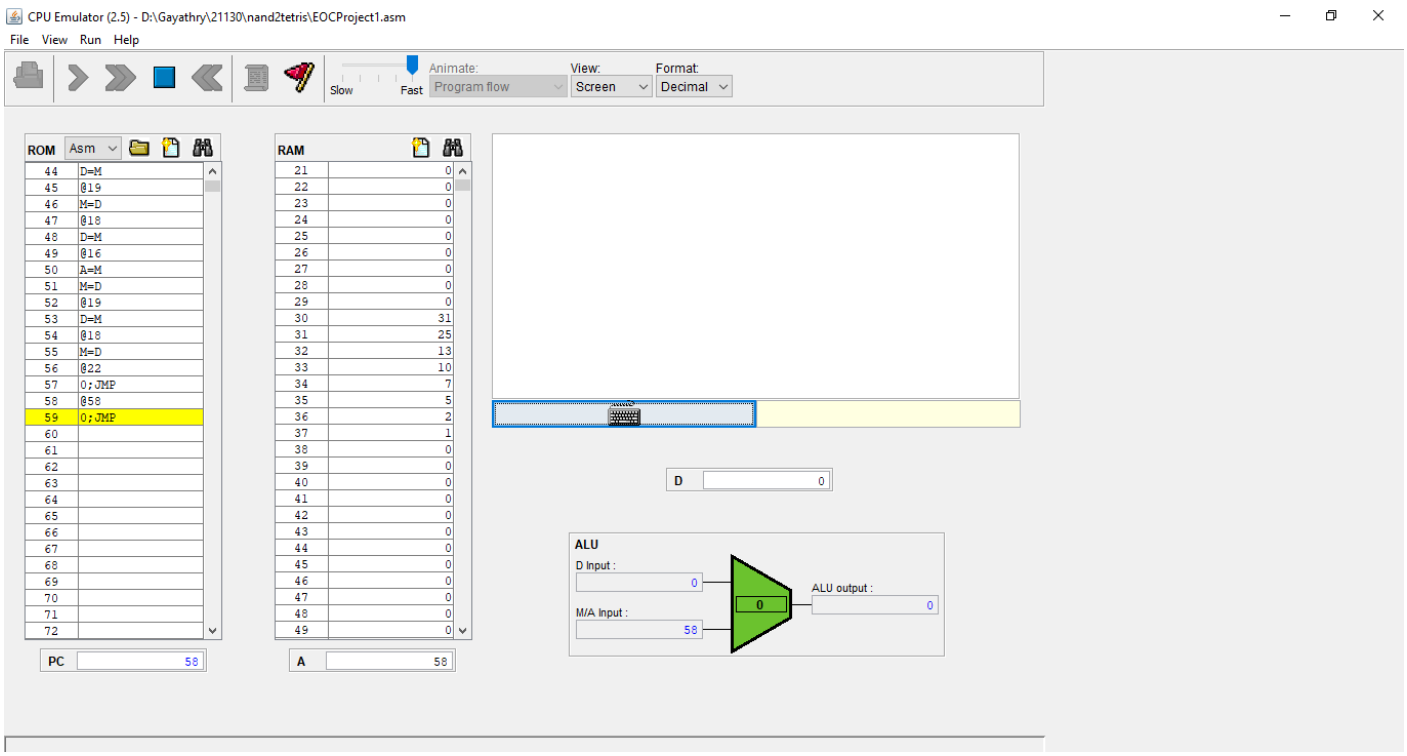
```
      @temp  // from temp to max
      D=M
      @max
      M=D

      @LOOP_TO
      0;JMP
(END)

      @END
      0;JMP
```

OUTPUT:

1. Inputted numbers from RAM[0] to RAM[7]



2. Storing numbers in RAM [30] to RAM [37] after sorting



INSIGHTS LEARNED ON HACK ASSEMBLY CODE:

- ➢ We observed how using loops helped the hack assembly code run more smoothly.
- ➢ Labels, variables, and the two fundamental types of instructions, A and C, were all comprehended.
- ➢ We got a better understanding of the Hack Instruction set and the three types of registers, A, D, and M, in order to write the code.
- ➢ We also learned how to run the asm code and get the appropriate output using the CPU Emulator.

CONTRIBUTION:

1. Anagha Viju - 20%
2. Annapoorna A K - 20%
3. Gayathry M Wariyar - 20%
4. Gouri S Vipin - 20%
5. Varalakshmi M P - 20%

****