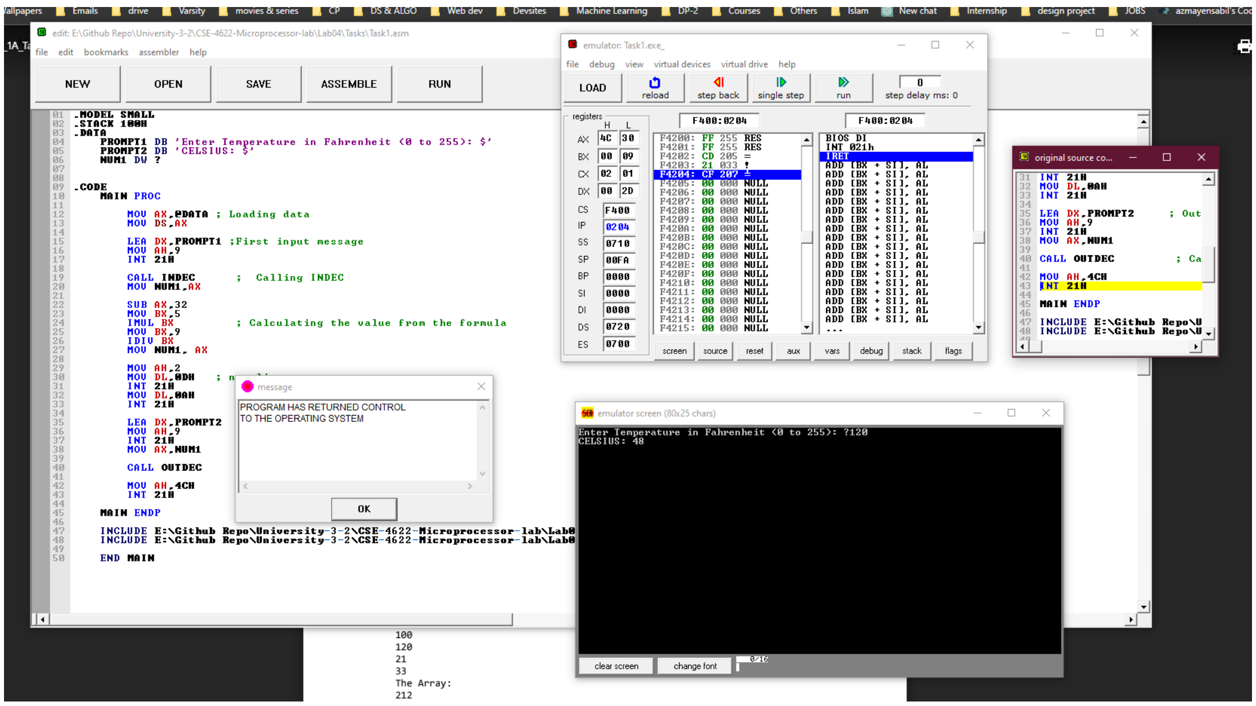
TASK-1

So the first task is to convert fahrenheit to celsius. We are already given a INDEC.asm and OUTDEC.asm file to convert the input value to decimal vice versa.



edit: E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\Tasks\Task1.asm

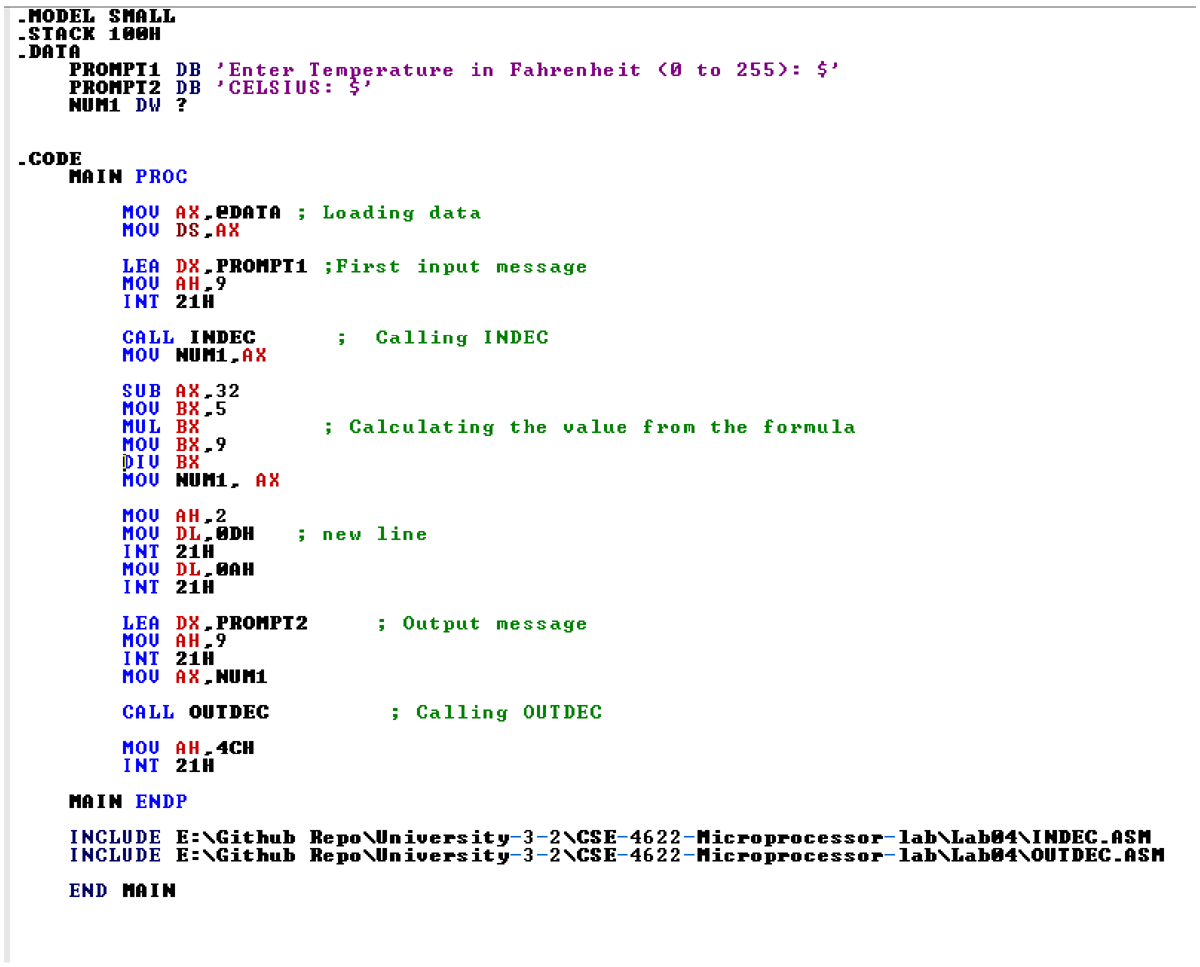
```
.MODEL SMALL
.STACK 100H
.DATA
    PROMPT1 DB 'Enter Temperature in Fahrenheit (0 to 255): $'
    PROMPT2 DB 'CELSIUS: $'
    NUM1 DW ?


.CODE
    MAIN PROC

        MOV AX,@DATA ; Loading data
        MOV DS,AX

        LEA DX,PROMPT1 ;First input message
        MOV AH,9
        INT 21H

        CALL INDEC      ;  Calling INDEC
        MOV NUM1,AX

        SUB AX,32
        MOV BX,5
        IMUL BX         ; Calculating the value from the formula
        MOV BX,9
        IDIV BX
        MOV NUM1, AX

        MOV AH,2
        MOV DL,0DH    ; n
        INT 21H
        MOV DL,0AH
        INT 21H

        LEA DX,PROMPT2
        MOV AH,9
        INT 21H
        MOV AX,NUM1

        CALL OUTDEC

        MOV AH,4CH
        INT 21H

    MAIN ENDP

    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab0
    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab0

    END MAIN
```

```
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
```

```
Enter Temperature in Fahrenheit (0 to 255): ?120
CELSIUS: 48
```

```
100
120
21
33
The Array:
212
```

This is the output. Now giving the snippet of the code.



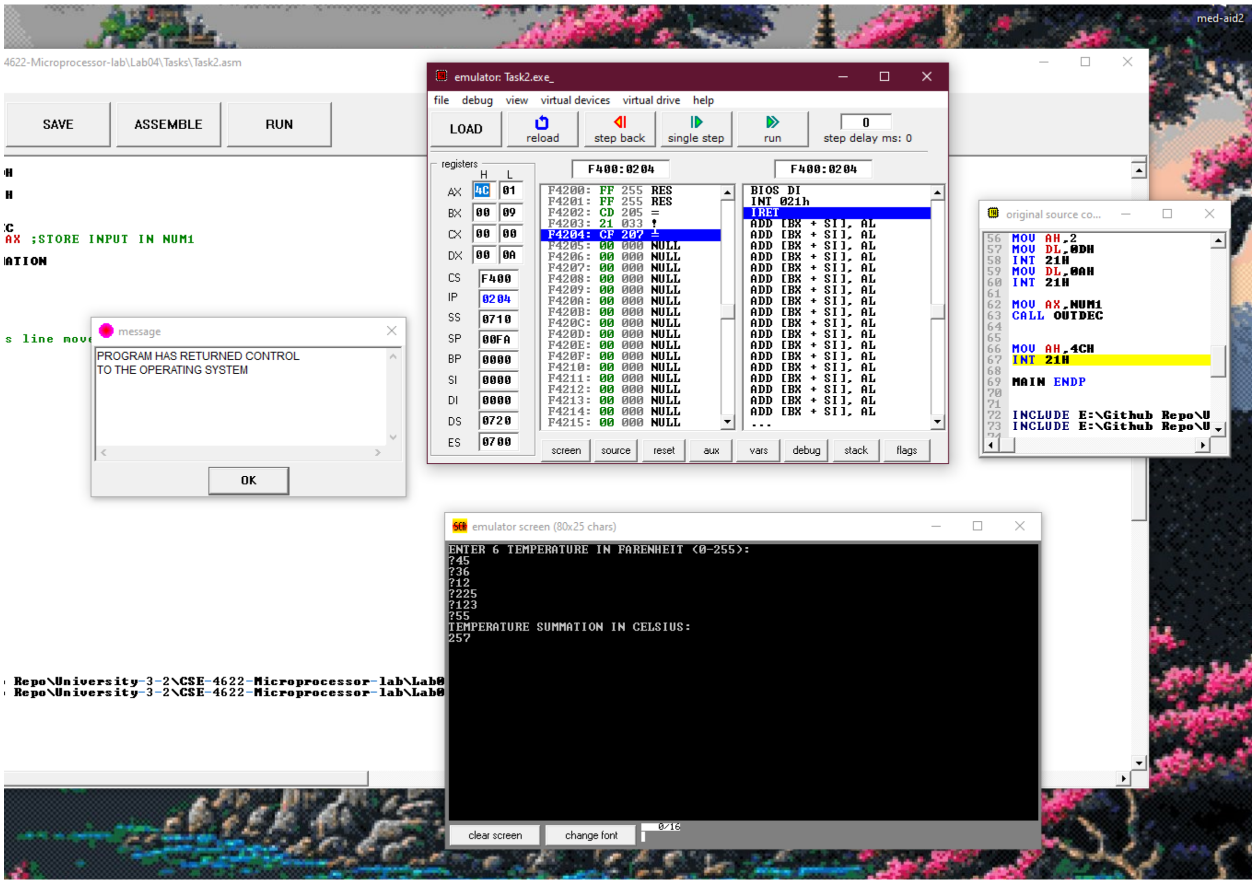
```
.MODEL SMALL
.STACK 100H
.DATA
    PROMPT1 DB 'Enter Temperature in Fahrenheit (0 to 255): $'
    PROMPT2 DB 'CELSIUS: $'
    NUM1 DW ?


.CODE
    MAIN PROC

        MOV AX,@DATA ; Loading data
        MOV DS,AX

        LEA DX,PROMPT1 ;First input message
        MOV AH,9
        INT 21H

        CALL INDEC      ;  Calling INDEC
        MOV NUM1,AX

        SUB AX,32
        MOV BX,5
        MUL BX         ; Calculating the value from the formula
        MOV BX,9
        DIV BX
        MOV NUM1, AX

        MOV AH,2
        MOV DL,0DH    ; new line
        INT 21H
        MOV DL,0AH
        INT 21H

        LEA DX,PROMPT2      ; Output message
        MOV AH,9
        INT 21H
        MOV AX,NUM1

        CALL OUTDEC          ; Calling OUTDEC

        MOV AH,4CH
        INT 21H

    MAIN ENDP

    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\INDEC.ASM
    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\OUTDEC.ASM

    END MAIN
```

So the steps are pretty simple. First we loaded necessary variables from the data segment. Then called the INDEC.asm to take input. Then executed the basic calculation to convert fahrenheit value to celsius value. Then again called the OUTDEC.asm to show the output.

TASK-2

This is almost similar to the previous one but here we are taking six inputs of Fahrenheit value and summing them up, after that converting that to celsius.

The result:



```
4622-Microprocessor-lab\Lab04\Tasks\Task2.asm

 AX ;STORE INPUT IN NUM1
 ATION

s line mov

 Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab0
 Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab0
```

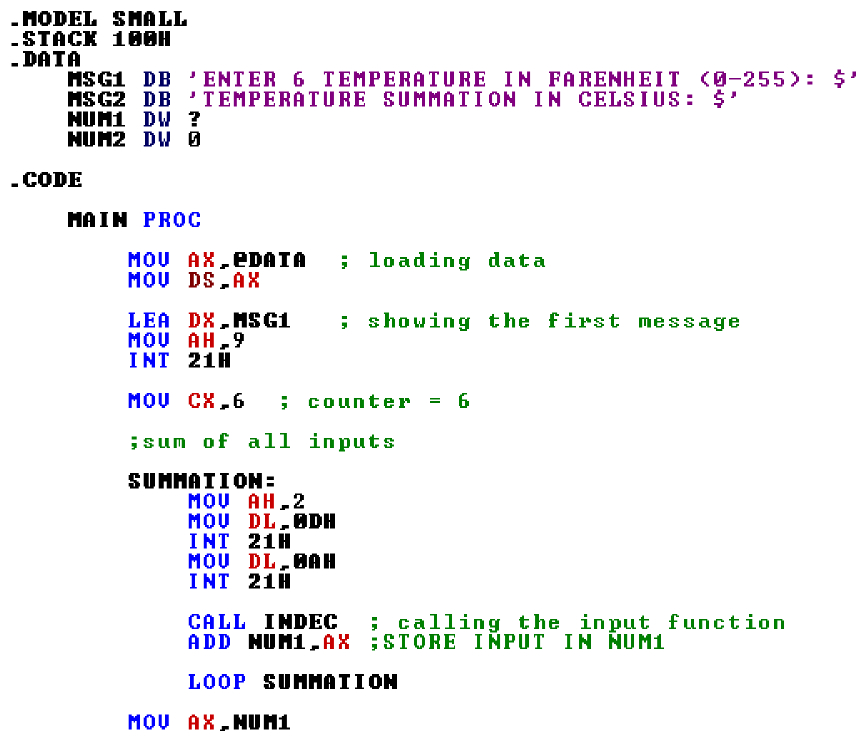
```
message
PROGRAM HAS RETURNED CONTROL
TO THE OPERATING SYSTEM
```

```
emulator: Task2.exe_

registers
     H   L
AX  4C  01
BX  00  09
CX  00  00
DX  00  0A
CS  F400
IP  0204
SS  0710
SP  00FA
BP  0000
SI  0000
DI  0000
DS  0720
ES  0700

F400:0204                  F400:0204
F4200: FF 255 RES          BIOS DI
F4201: FF 255 RES          INT 021h
F4202: CD 205 =            IRET
F4203: 21 033 !            ADD [BX + SI], AL
F4204: CF 207 ±            ADD [BX + SI], AL
F4205: 00 000 NULL         ADD [BX + SI], AL
F4206: 00 000 NULL         ADD [BX + SI], AL
F4207: 00 000 NULL         ADD [BX + SI], AL
F4208: 00 000 NULL         ADD [BX + SI], AL
F4209: 00 000 NULL         ADD [BX + SI], AL
F420A: 00 000 NULL         ADD [BX + SI], AL
F420B: 00 000 NULL         ADD [BX + SI], AL
F420C: 00 000 NULL         ADD [BX + SI], AL
F420D: 00 000 NULL         ADD [BX + SI], AL
F420E: 00 000 NULL         ADD [BX + SI], AL
F420F: 00 000 NULL         ADD [BX + SI], AL
F4210: 00 000 NULL         ADD [BX + SI], AL
F4211: 00 000 NULL         ADD [BX + SI], AL
F4212: 00 000 NULL         ADD [BX + SI], AL
F4213: 00 000 NULL         ADD [BX + SI], AL
F4214: 00 000 NULL         ADD [BX + SI], AL
F4215: 00 000 NULL         ...
```

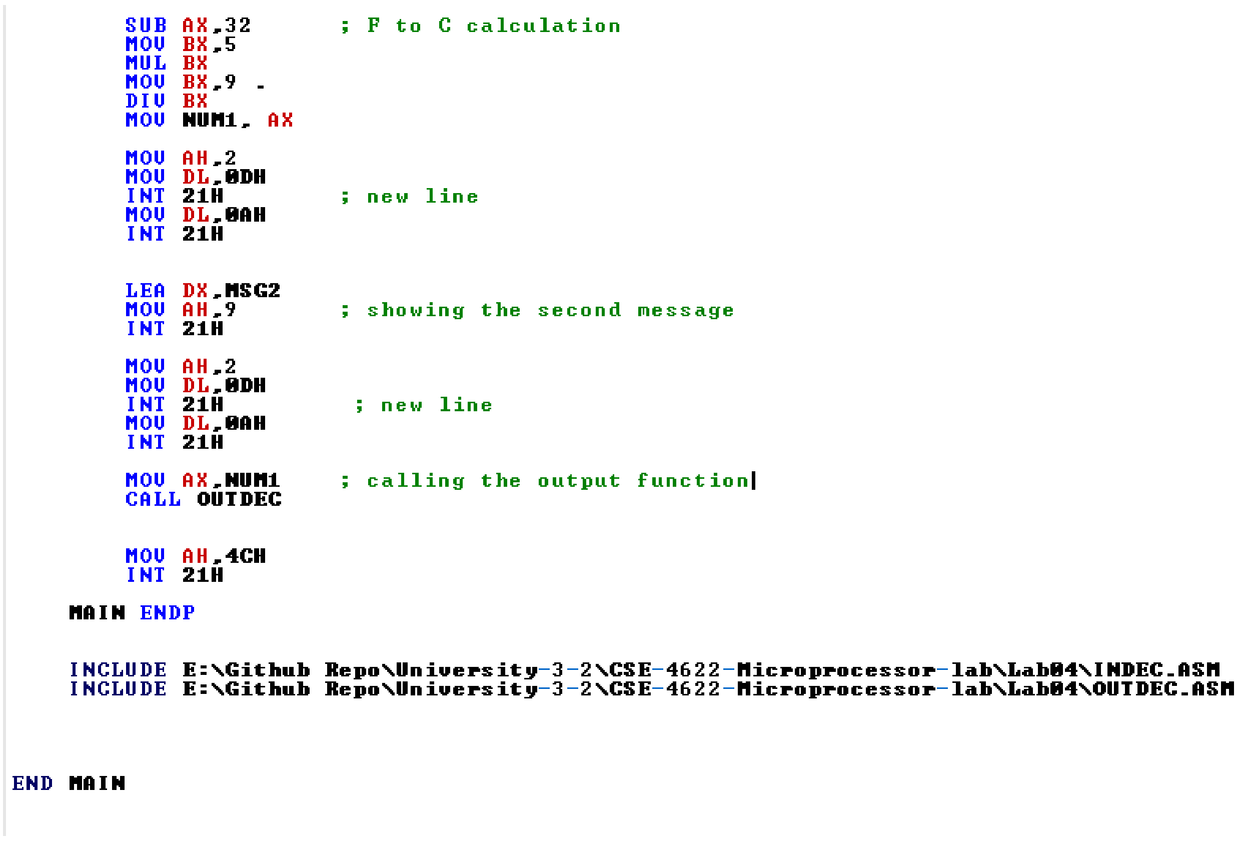
```
original source co...
MOV AH,2
MOV DL,0DH
INT 21H
MOV DL,0AH
INT 21H

MOV AX,NUM1
CALL OUTDEC


MOV AH,4CH
INT 21H

MAIN ENDP


INCLUDE E:\Github Repo\U
INCLUDE E:\Github Repo\U
```

```
emulator screen (80x25 chars)
ENTER 6 TEMPERATURE IN FARENHEIT (0-255):
?45
?36
?12
?225
?123
?55
TEMPERATURE SUMMATION IN CELSIUS:
257
```

The code snippet is given below:



```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1 DB 'ENTER 6 TEMPERATURE IN FARENHEIT (0-255): $'
    MSG2 DB 'TEMPERATURE SUMMATION IN CELSIUS: $'
    NUM1 DW ?
    NUM2 DW 0

.CODE

    MAIN PROC

        MOV AX,@DATA   ; loading data
        MOV DS,AX

        LEA DX,MSG1    ; showing the first message
        MOV AH,9
        INT 21H

        MOV CX,6   ; counter = 6

        ;sum of all inputs

        SUMMATION:
            MOV AH,2
            MOV DL,0DH
            INT 21H
            MOV DL,0AH
            INT 21H

            CALL INDEC  ; calling the input function
            ADD NUM1,AX ;STORE INPUT IN NUM1

            LOOP SUMMATION

        MOV AX,NUM1
```

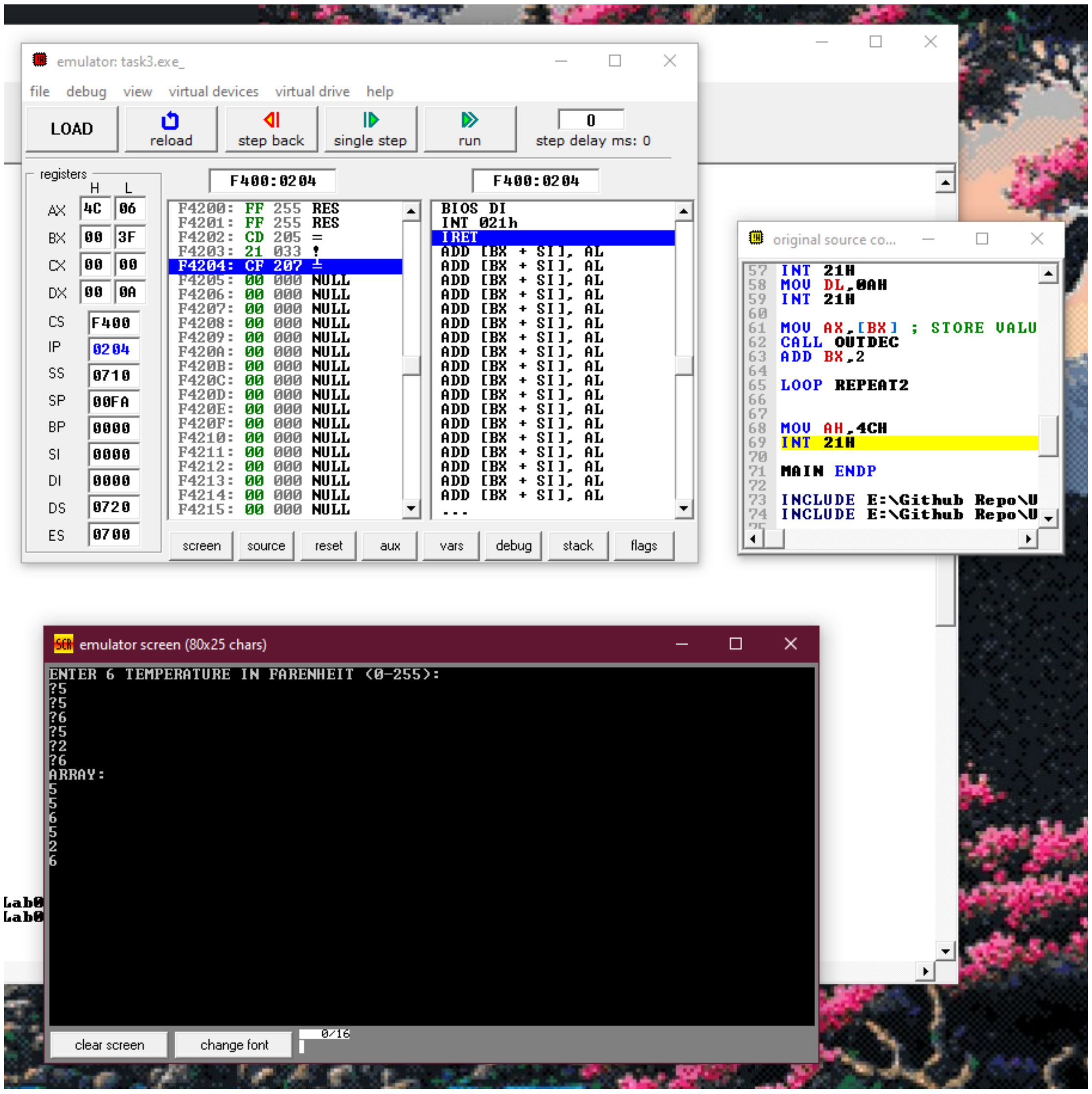


```
        SUB AX,32       ; F to C calculation
        MOV BX,5
        MUL BX
        MOV BX,9 .
        DIV BX
        MOV NUM1, AX

        MOV AH,2
        MOV DL,0DH
        INT 21H         ; new line
        MOV DL,0AH
        INT 21H


        LEA DX,MSG2
        MOV AH,9        ; showing the second message
        INT 21H

        MOV AH,2
        MOV DL,0DH
        INT 21H          ; new line
        MOV DL,0AH
        INT 21H

        MOV AX,NUM1     ; calling the output function
        CALL OUTDEC


        MOV AH,4CH
        INT 21H

    MAIN ENDP


    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\INDEC.ASM
    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\OUTDEC.ASM



END MAIN
```

So here we just used a loop to call the INDEC.asm to take input and every time we took the input, we also did the summation part and stored it inside the num1 variable. Then the rest of the process is very similar to the previous one.

TASK-3

In this task, our job is to take 6 inputs and store them inside an array. Then display them in the console.



```
      H   L
AX   4C  06
BX   00  3F
CX   00  00
DX   00  0A
CS   F400
IP   0204
SS   0710
SP   00FA
BP   0000
SI   0000
DI   0000
DS   0720
ES   0700
```

```
F400:0204
F4200: FF 255 RES
F4201: FF 255 RES
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ╧
F4205: 00 000 NULL
F4206: 00 000 NULL
F4207: 00 000 NULL
F4208: 00 000 NULL
F4209: 00 000 NULL
F420A: 00 000 NULL
F420B: 00 000 NULL
F420C: 00 000 NULL
F420D: 00 000 NULL
F420E: 00 000 NULL
F420F: 00 000 NULL
F4210: 00 000 NULL
F4211: 00 000 NULL
F4212: 00 000 NULL
F4213: 00 000 NULL
F4214: 00 000 NULL
F4215: 00 000 NULL
```

```
F400:0204
BIOS DI
INT 021h
IRET
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
ADD [BX + SI], AL
...
```

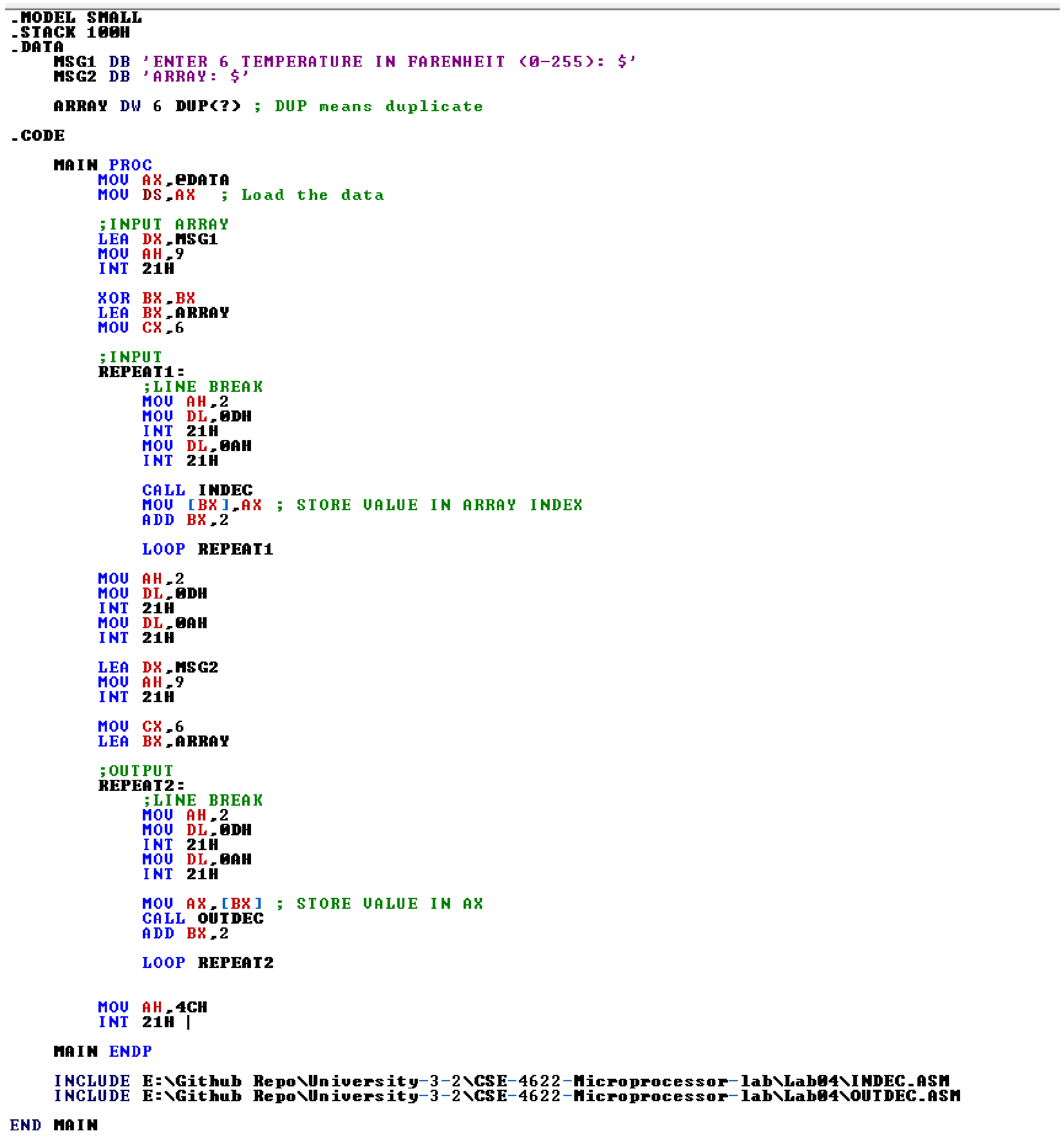
```
INT 21H
MOV DL,0AH
INT 21H

MOV AX,[BX] ; STORE VALU
CALL OUTDEC
ADD BX,2

LOOP REPEAT2


MOV AH,4CH
INT 21H

MAIN ENDP

INCLUDE E:\Github Repo\U
INCLUDE E:\Github Repo\U
```

```
ENTER 6 TEMPERATURE IN FARENHEIT (0-255):
?5
?5
?6
?5
?2
?6
ARRAY:
5
5
6
5
2
6
```

```
Lab0
Lab0
```

And the code is kind of similar to the previous ones but only difference is that we had to store it in an array.

The code:



```
.MODEL SMALL
.STACK 100H
.DATA
    MSG1 DB 'ENTER 6 TEMPERATURE IN FARENHEIT (0-255): $'
    MSG2 DB 'ARRAY: $'

    ARRAY DW 6 DUP(?) ; DUP means duplicate

.CODE

    MAIN PROC
        MOV AX,@DATA
        MOV DS,AX   ; Load the data

        ;INPUT ARRAY
        LEA DX,MSG1
        MOV AH,9
        INT 21H

        XOR BX,BX
        LEA BX,ARRAY
        MOV CX,6

        ;INPUT
        REPEAT1:
            ;LINE BREAK
            MOV AH,2
            MOV DL,0DH
            INT 21H
            MOV DL,0AH
            INT 21H

            CALL INDEC
            MOV [BX],AX ; STORE VALUE IN ARRAY INDEX
            ADD BX,2

            LOOP REPEAT1

        MOV AH,2
        MOV DL,0DH
        INT 21H
        MOV DL,0AH
        INT 21H

        LEA DX,MSG2
        MOV AH,9
        INT 21H

        MOV CX,6
        LEA BX,ARRAY

        ;OUTPUT
        REPEAT2:
            ;LINE BREAK
            MOV AH,2
            MOV DL,0DH
            INT 21H
            MOV DL,0AH
            INT 21H

            MOV AX,[BX] ; STORE VALUE IN AX
            CALL OUTDEC
            ADD BX,2

            LOOP REPEAT2


        MOV AH,4CH
        INT 21H

    MAIN ENDP

    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\INDEC.ASM
    INCLUDE E:\Github Repo\University-3-2\CSE-4622-Microprocessor-lab\Lab04\OUTDEC.ASM

END MAIN
```