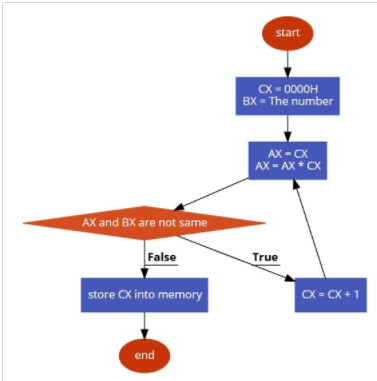


1. SQRT:



Code

**Assume CS: Code DS: Data**

**DATA SEGMENT**

**NUM DW 211H**

**SQRT DW ?**

**DATA ENDS**

**CODE SEGMENT**

**START:**

**MOV AX, @DATA**

**MOV DS, AX**

**MOV CX,0000H**

**MOV BX, Num**

**L1:MOV AX,CX**

**MUL CX**

**CMP AX, BX**

**JZ STORE**

**INC CX**

**JNZ L1**

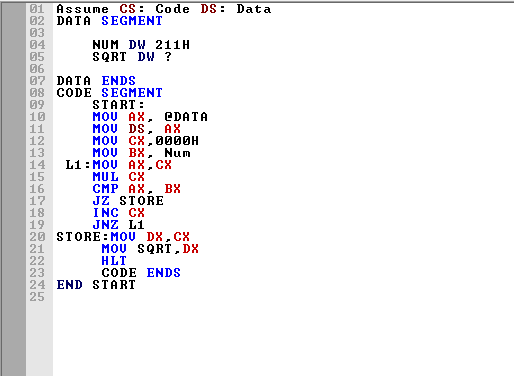
**STORE:MOV DX,CX**

**MOV SQRT,DX**

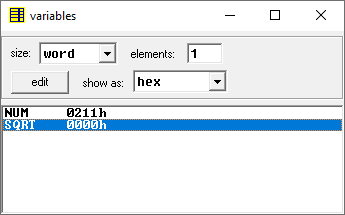
**HLT**

**CODE ENDS**

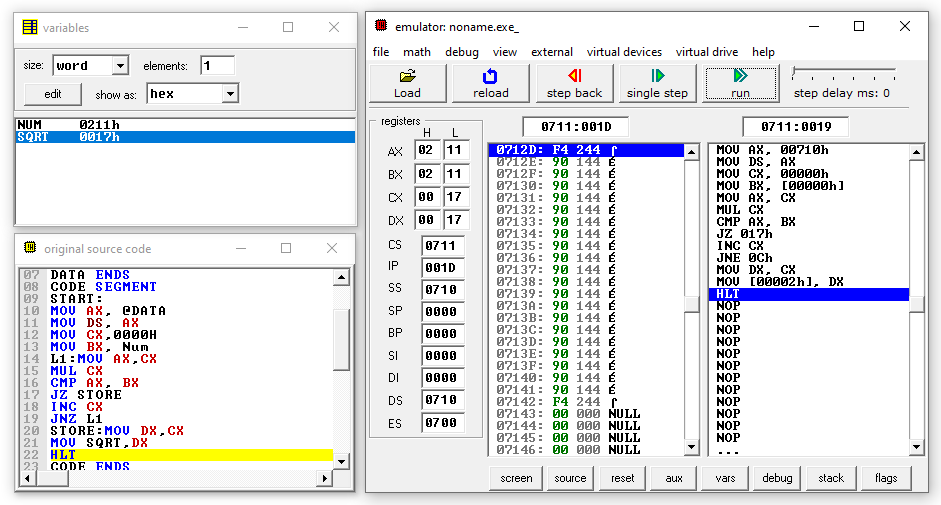
**END START**

****

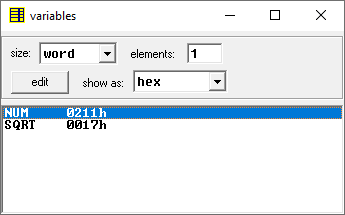
**Before Exe**

****

**OUTPUT:**

****

**After Exe**

****

2)GP

**Code:**

**Assume CS: Code DS: Data**

**DATA SEGMENT**

**Input DW 0500H**

**Output DW 0600H**

**DATA ENDS**

**CODE SEGMENT**

**START:**

**MOV AX, @DATA**

**MOV DS, AX**

**MOV SI, Input**

**MOV CL, [SI]**

**MOV CH, 00H**

**INC SI**

**MOV AL, [SI]**

**MOV AH, 00H**

**MOV DI, Output**

**MOV [DI], AL**

**INC DI**

**DEC CL**

**INC SI**

**MOV BL, [SI]**

**L1:MUL BL**

**MOV [DI], AL**

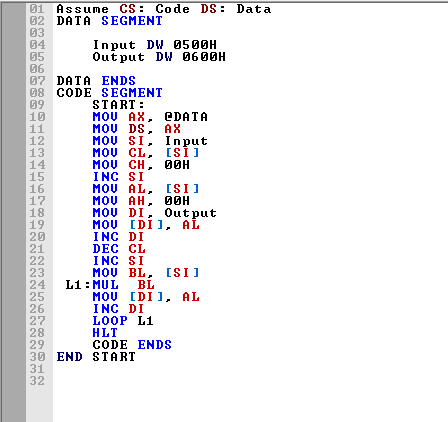
**INC DI**

**LOOP L1**

**HLT**

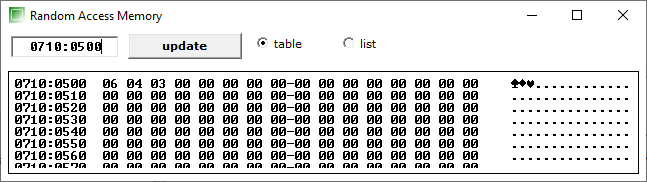
**CODE ENDS**

**END START**

****

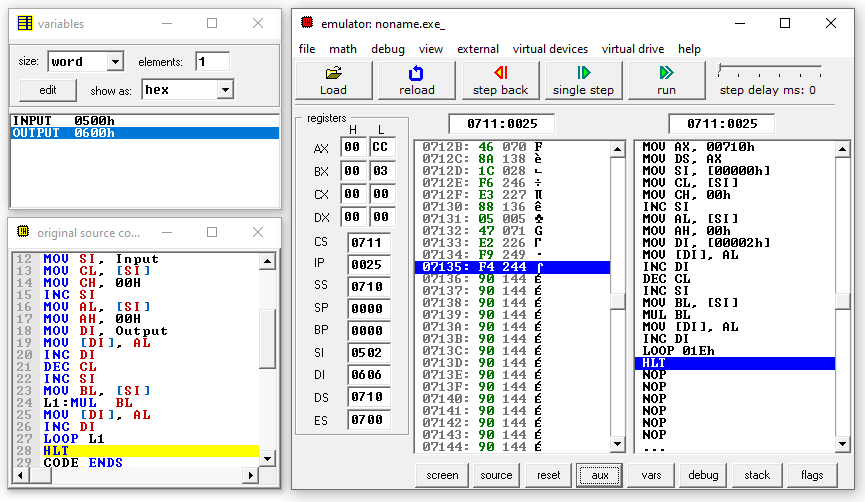
**Input:**

**Before Execution**

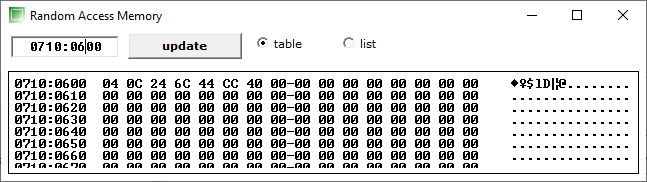
****

**Given input (Number of terms) n=06h in 0710:0500h and first term a=04h in 0710:0501h and common ratio r =03h in 0710:0502h**

**Output:**

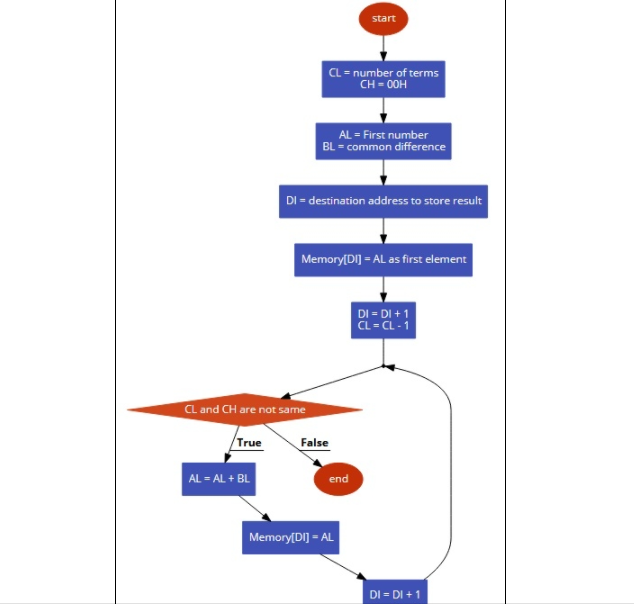
****

**After Execution:**

****

**GP Series obtained starting from 600 location**

3)AP

****

**CODE:**

**Assume CS: Code DS: Data**

**DATA SEGMENT**

**Input DW 0500H**

**Output DW 0600H**

**DATA ENDS**

**CODE SEGMENT**

**START:**

**MOV AX, @DATA**

**MOV DS, AX**

**MOV SI, Input**

**MOV CL, [SI]**

**MOV CH, 00H**

**INC SI**

**MOV AL, [SI]**

**MOV AH, 00H**

**MOV DI, Output**

**MOV [DI], AL**

**INC DI**

**DEC CL**

**INC SI**

**MOV BL, [SI]**

**L1:ADD AL, BL**

**MOV [DI], AL**

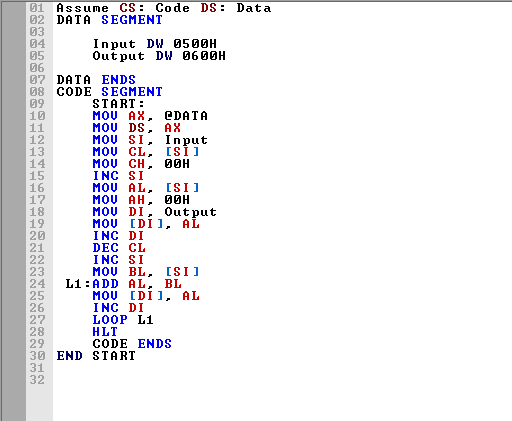
**INC DI**

**LOOP L1**

**HLT**

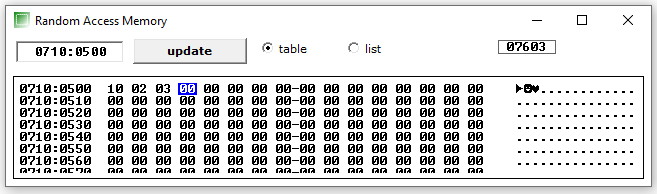
**CODE ENDS**

**END START**

****

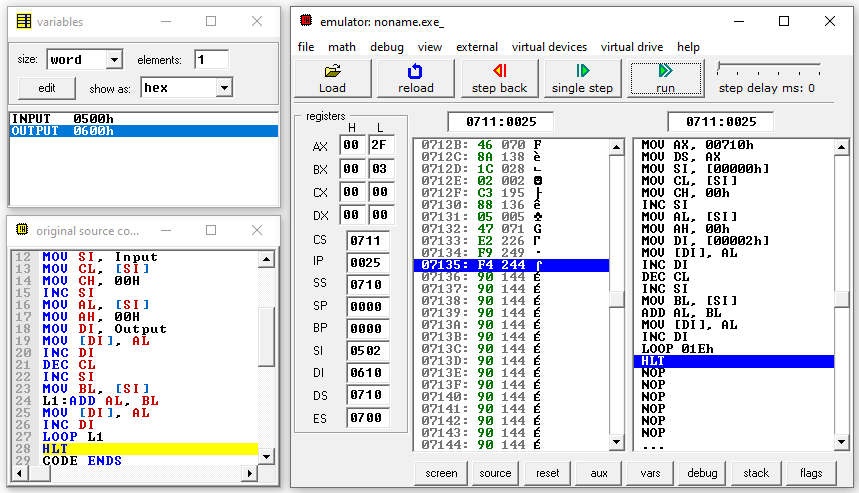
**Input:**

**Before Execution**

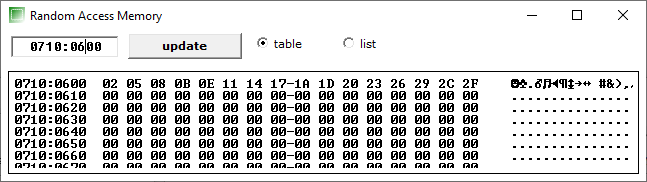
****

**Given input Number of Terms n=10h in 0710:0500h and first term a=02h in 0710:0501h and common difference d=03h in 0710:0502h**

**Output:**

****

**After Execution:**

****

**AP Series obtain starting from 600 location.**

4)Addition and Average of a series

**Code**

**Assume CS: Code DS: Data**

**DATA SEGMENT**

**InputLoc DW 0500H**

**SIZE DB 6h**

**SUM DW ?**

**AVG DW ?**

**DATA ENDS**

**CODE SEGMENT**

**START:**

**MOV AX, @DATA**

**MOV DS, AX**

**MOV SI, InputLoc**

**MOV CL, size**

**MOV CH, 00H**

**MOV AL, [SI]**

**MOV AH, 00H**

**DEC CL**

**INC SI**

**L1:MOV BL,[SI]**

**MOV BH,00H**

**ADD AX,BX**

**INC SI**

**LOOP L1**

**MOV SUM,AX**

**MOV SUM,AX**

**MOV BL, SIZE**

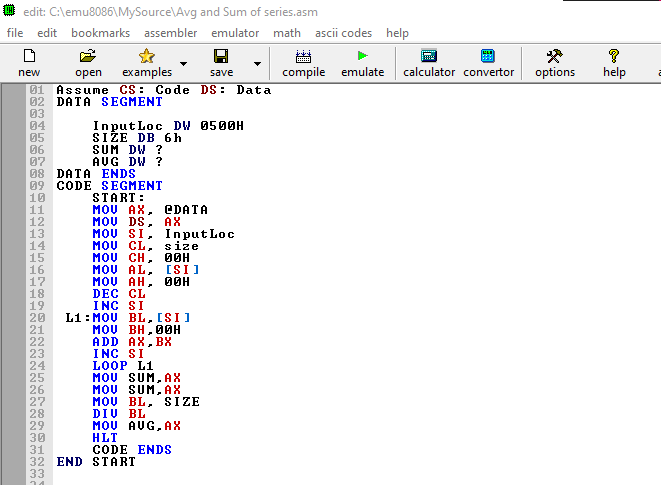
**DIV BL**

**MOV AVG,AX**

**HLT**

**CODE ENDS**

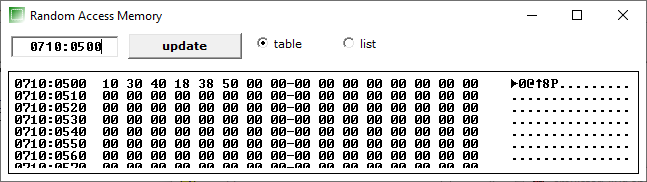
**END START**

****

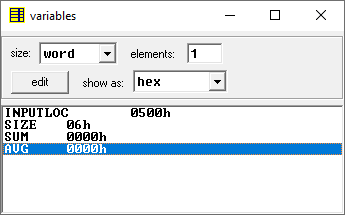
**Input:**

**Before Execution**

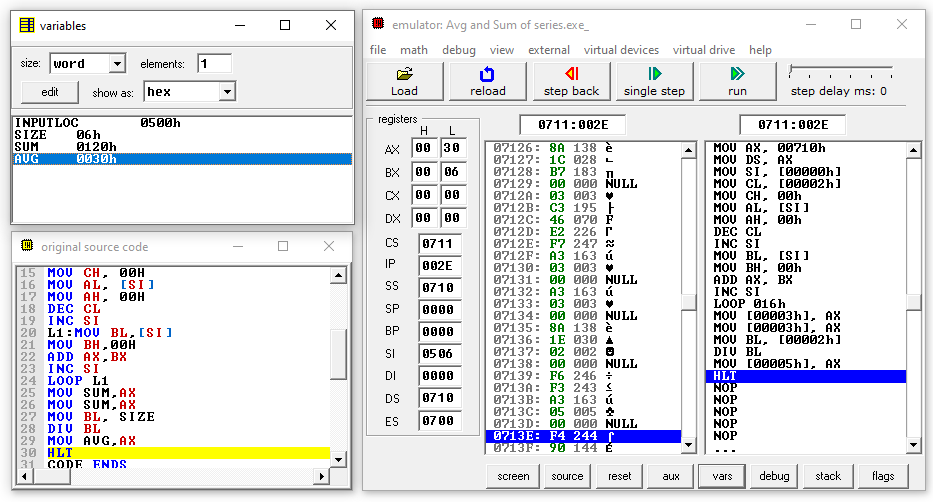
**Memory**

****

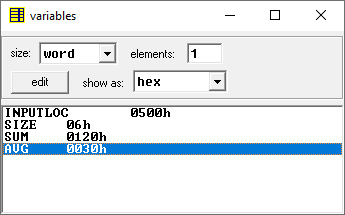
**Variables**

****

**OUTPUT**

****

**Variables**

****

5)Descending

**🡪 We can follow the same Procedure as in Ascending order**

**Only 1 line of code will change i.e.**

**In Ascending code**

**CMP AL,BL**

**In Descending code**

**CMP BL,AL**

**CODE**

**DATA SEGMENT**

**Array DB 57H,10H,56H,25H,32H,35H**

**SIZE DB 6H**

**DATA ENDS**

**CODE SEGMENT**

**ASSUME CS:CODE,DS:DATA**

**START: MOV AX,DATA**

**MOV DS,AX**

**MOV CH,SIZE**

**DEC CH**

**UP2: MOV CL,SIZE**

**DEC CL**

**LEA SI,Array**

**UP1: MOV AL,[SI]**

**MOV BL,[SI+1]**

**CMP BL,AL**

**JC DOWN**

**MOV DL,[SI+1]**

**XCHG [SI],DL**

**MOV [SI+1],DL**

**DOWN: INC SI**

**DEC CL**

**JNZ UP1**

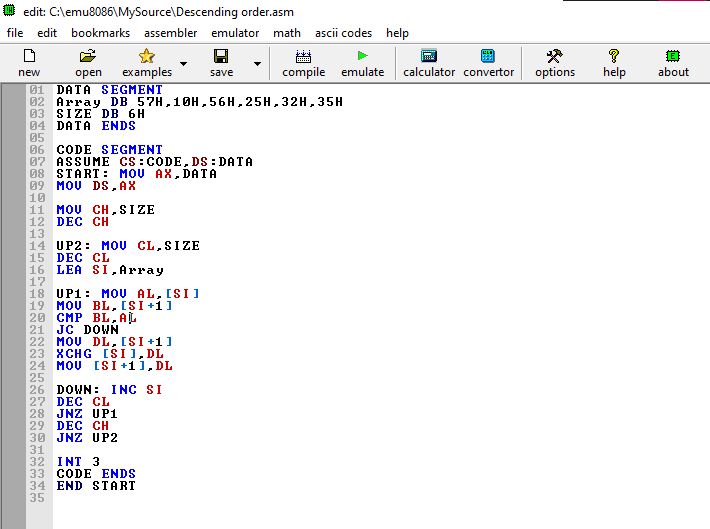
**DEC CH**

**JNZ UP2**

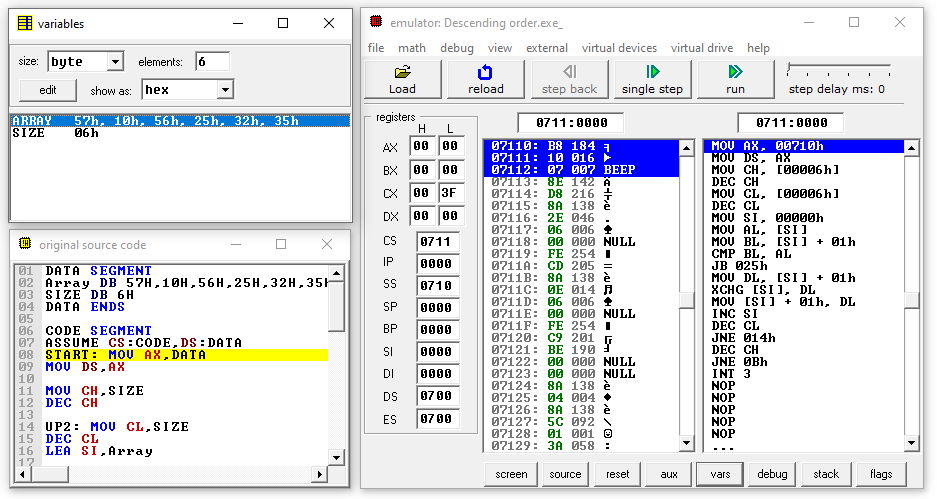
**INT 3**

**CODE ENDS**

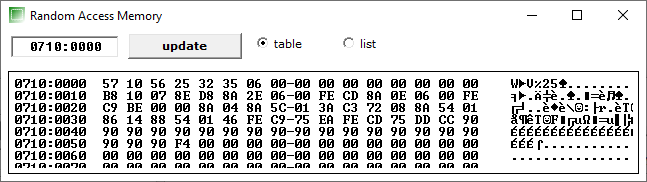
**END START**

****

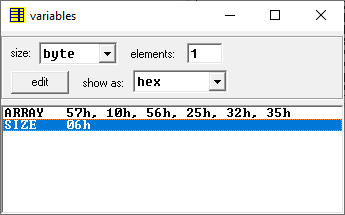
**Before Execution:**

****

**Memory Location**

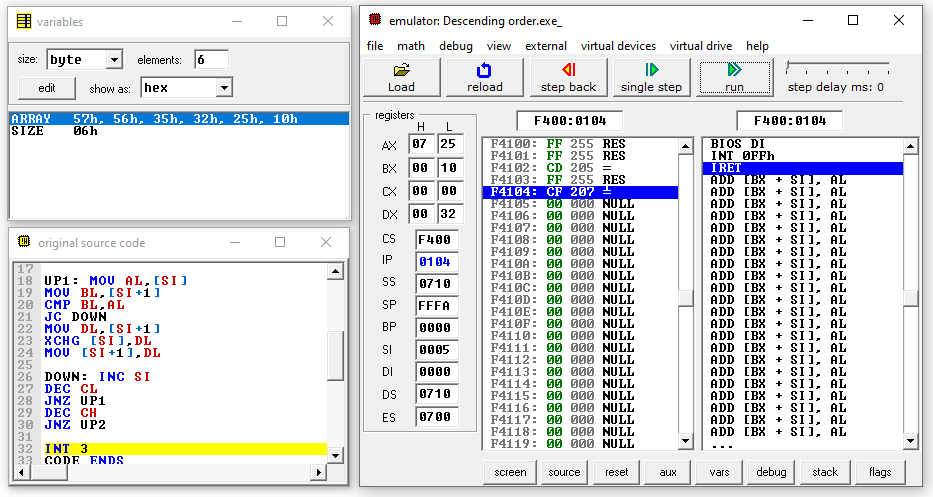
****

**Variables**

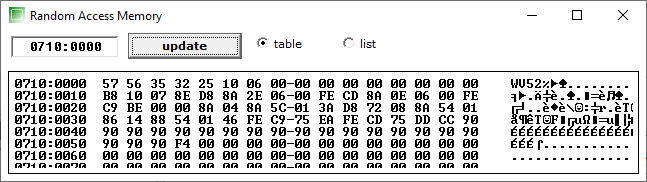
****

**OUTPUT:**

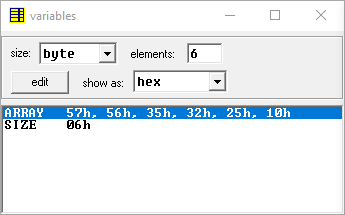
**After Execution:**

****

**Memory Location**

****

**Variables**

****

6)Ascending

**Copy Algo, Flowchart, design from the lab doc change Calculations an below SS**

**Lab DA6 🡪Experiment 8.1 🡪Question 1**

**CODE**

**DATA SEGMENT**

**Array DB 57H,10H,56H,25H,32H,35H**

**SIZE DB 6H**

**DATA ENDS**

**CODE SEGMENT**

**ASSUME CS:CODE,DS:DATA**

**START: MOV AX,DATA**

**MOV DS,AX**

**MOV CH,SIZE**

**DEC CH**

**UP2: MOV CL,SIZE**

**DEC CL**

**LEA SI,Array**

**UP1: MOV AL,[SI]**

**MOV BL,[SI+1]**

**CMP AL,BL**

**JC DOWN**

**MOV DL,[SI+1]**

**XCHG [SI],DL**

**MOV [SI+1],DL**

**DOWN: INC SI**

**DEC CL**

**JNZ UP1**

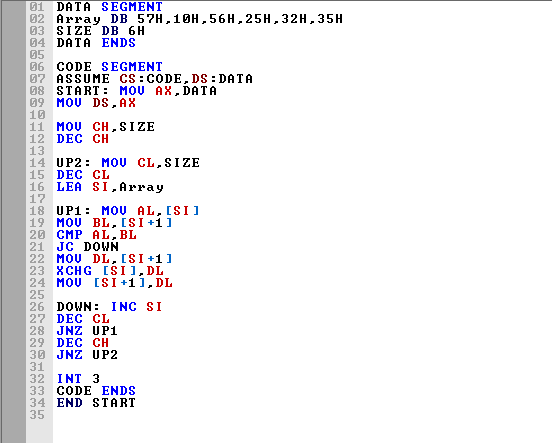
**DEC CH**

**JNZ UP2**

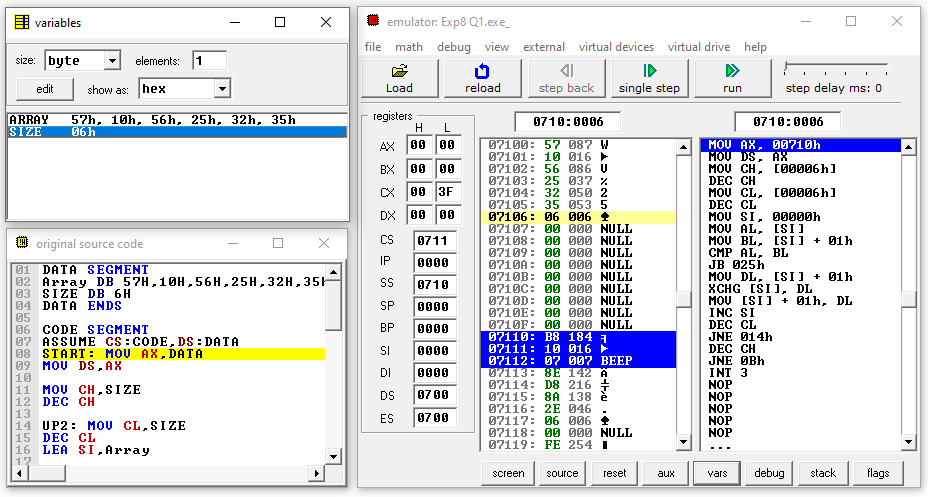
**INT 3**

**CODE ENDS**

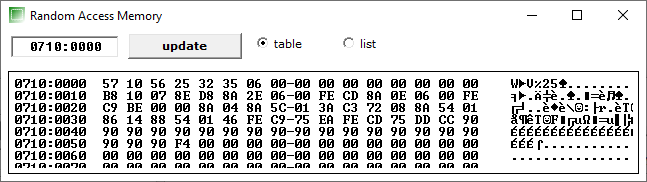
**END START**

****

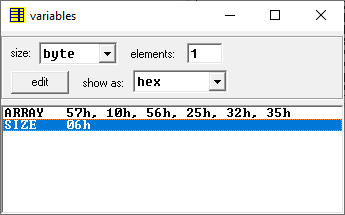
**Before Execution:**

****

**Memory Location**

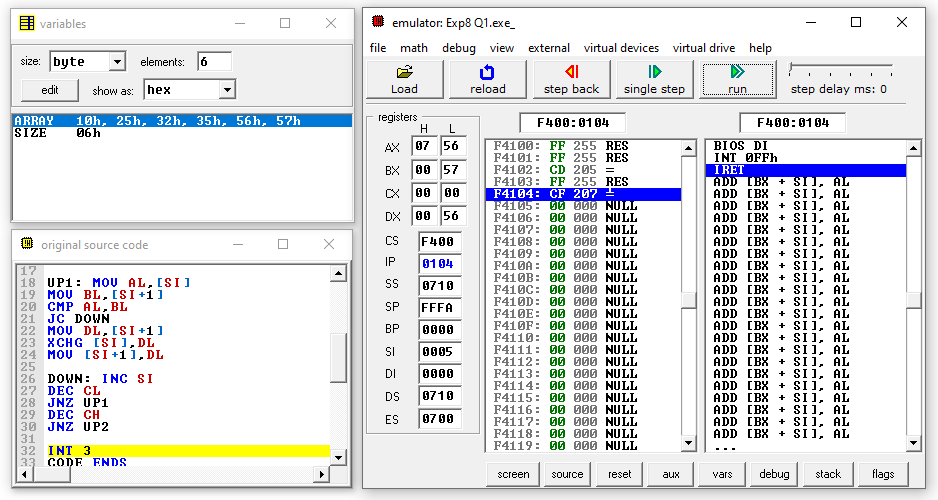
****

**Variables**

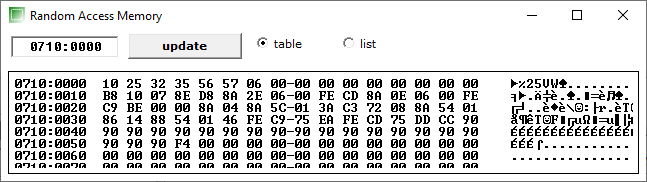
****

**OUTPUT:**

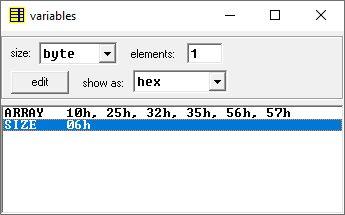
**After Execution:**

****

**Memory Location**

****

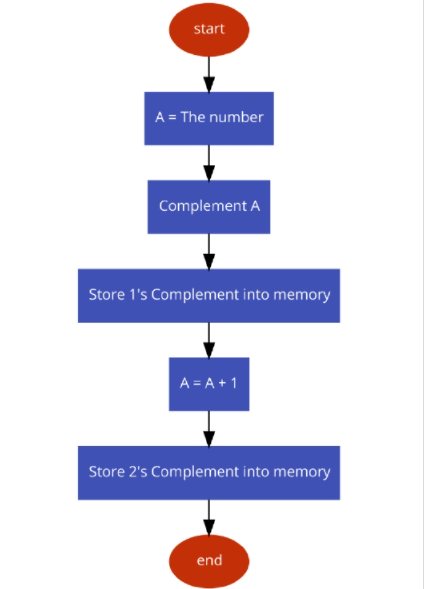
**Variables**

****

7)Fibanocci series

**Lab DA5 🡪 Experiment 6 🡪 Question 2 🡪2nd part**

8 a)Binary to 2’s comp



**Code**

**Data Segment**

**bin db 00000010B**

**2comp db ?**

**Data Ends**

**Code Segment**

**Assume cs:code, ds:data**

**Begin:**

**mov ax, data**

**mov ds, ax**

**mov ah, 0000h**

**mov al, bin**

**NOT al**

**mov bl, al**

**adc al, 00000001B**

**mov bl, al**

**Exit:**

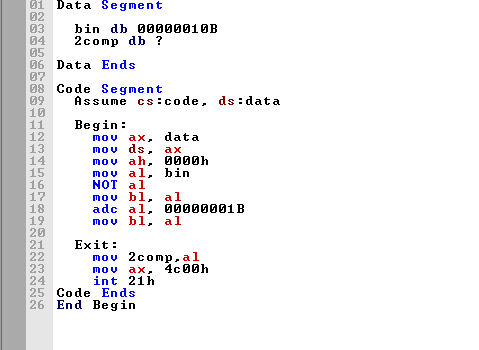
**mov 2comp,al**

**mov ax, 4c00h**

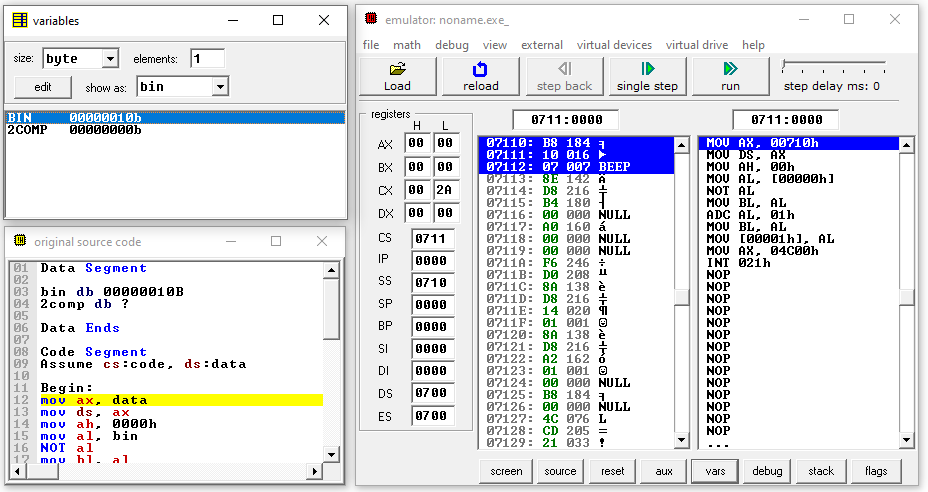
**int 21h**

**Code Ends**

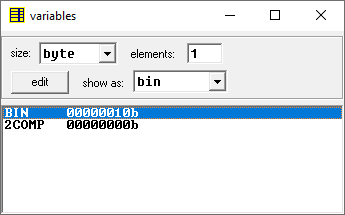
**End Begin**

****

**Before Exe**

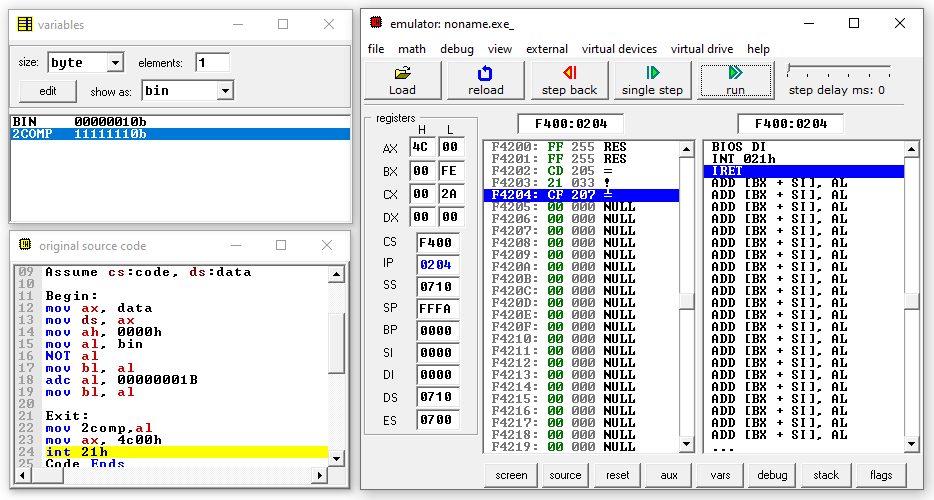
****

**Variables**

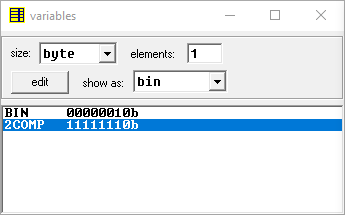
****

**OUTPUT**

**After Exe**



Variables:

****

8 b)BCD to binary

**Lab DA6 🡪Experiment7 🡪 Question2**

8 c)Binary to BCD

**Lab DA6 🡪Experiment7 🡪 Question1**

9) Factorial of number whose input is in memory

**Lab DA5 🡪Exp 6🡪 Question2 🡪 Part1**

**(Change Memory Location according to Question)**

**10) Largest and Smallest numbers in Array**

**Code**

**data segment**

**array db 67h,25h,14h,02h**

**size db 04h**

**small db 0h**

**large db 0h**

**data ends**

**code segment**

**start:**

**mov ax,data**

**mov ds,ax**

**mov cl, size**

**mov si , offset array**

**mov al,[si]**

**dec cl**

**up1:inc si**

**cmp al,[si]**

**jc dn1**

**mov al,[si]**

**jc dn1**

**dn1:dec cl**

**jnz up1**

**mov cl, size**

**mov si , offset array**

**mov bl,[si]**

**dec cl**

**up2:inc si**

**cmp [si], bl**

**jc dn2**

**mov bl,[si]**

**jc dn2**

**dn2:dec cl**

**jnz up2**

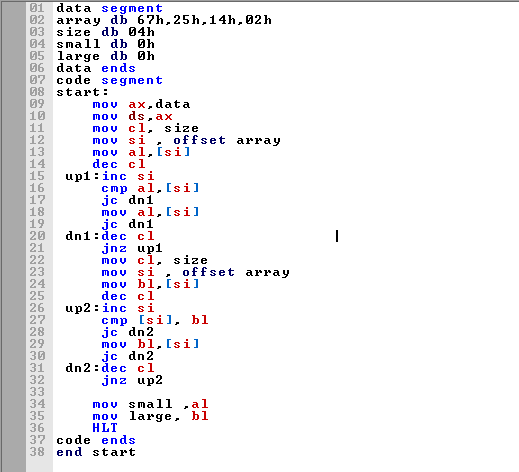
**mov small ,al**

**mov large, bl**

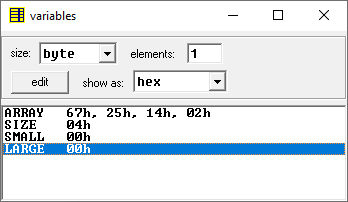
**HLT**

**code ends**

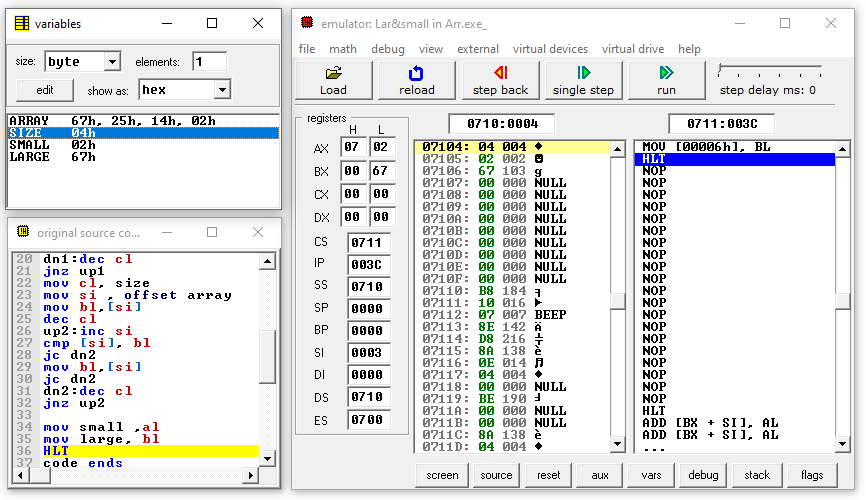
**end start**

****

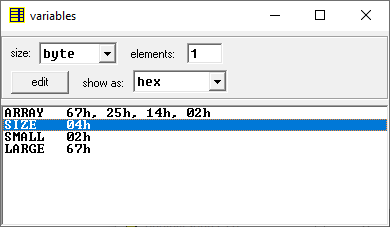
**Variables Before Execution**

****

**OUTPUT**

****

**Variables After Execution**

****

**11)**

a) Calculate Length of string

**CODE**

**;macro for printing a string**

**print macro m**

**mov ah,09h**

**mov dx,offset m**

**int 21h**

**endm**

**.model small**

**.data**

**msg db 10,13, "Exiting the program $"**

**empty db 10,13, " $"**

**str1 db 25,?,25 dup('$')**

**len db ?**

**mstring db 10,13, "Enter the string: $"**

**mlength db 10,13, "Length is: $"**

**;\*\*\*\*\*\*\*\*\*\* Code Segment \*\*\*\*\*\*\*\*\*\*\*\***

**.code**

**start:**

**mov ax,@data**

**mov ds,ax**

**print mstring**

**call accept\_string ;function call to accept a string**

**mov cl,str1+1 ;storing length in cl from first byte of the array**

**mov bl,cl ;copying in bl for displaying**

**print mlength**

**call display1 ;printing the length**

**exit:**

**mov ah,4ch ;exit the program**

**int 21h**

**;accept procedure**

**accept proc near**

**mov ah,01**

**int 21h**

**ret**

**accept endp**

**display1 proc near**

**mov al,bl**

**mov bl,al**

**and al,0f0h**

**mov cl,04**

**rol al,cl**

**cmp al,09**

**jbe number**

**add al,07**

**number: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**mov al,bl**

**and al,00fh**

**cmp al,09**

**jbe number2**

**add al,07**

**number2: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**ret**

**display1 endp**

**accept\_string proc near**

**mov ah,0ah ;accept string from user function**

**mov dx,offset str1 ; store the string in memory pointed by "DX"**

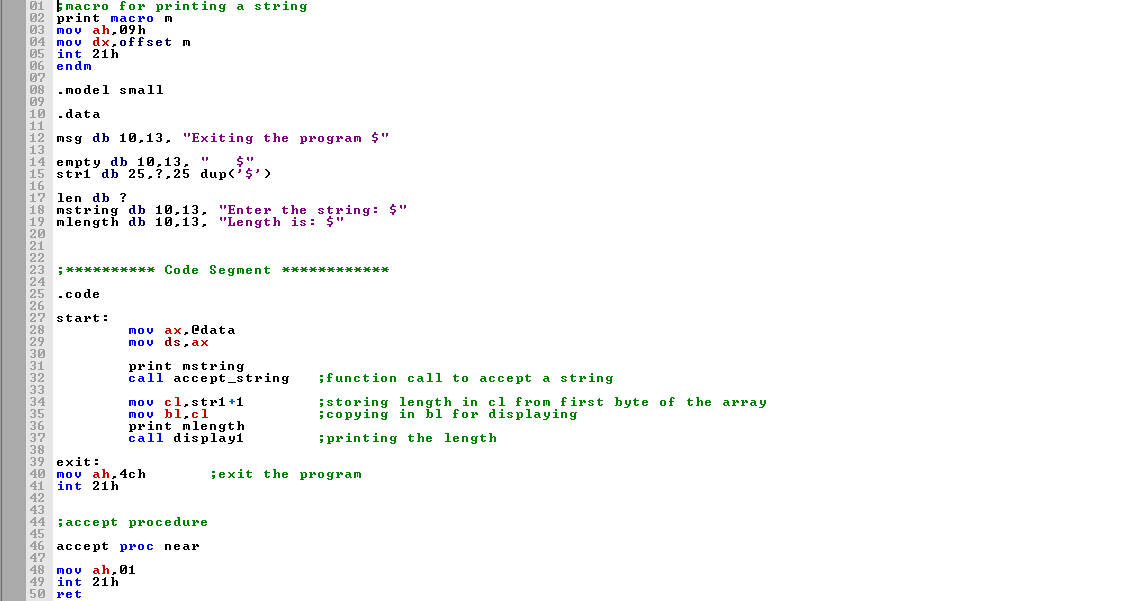
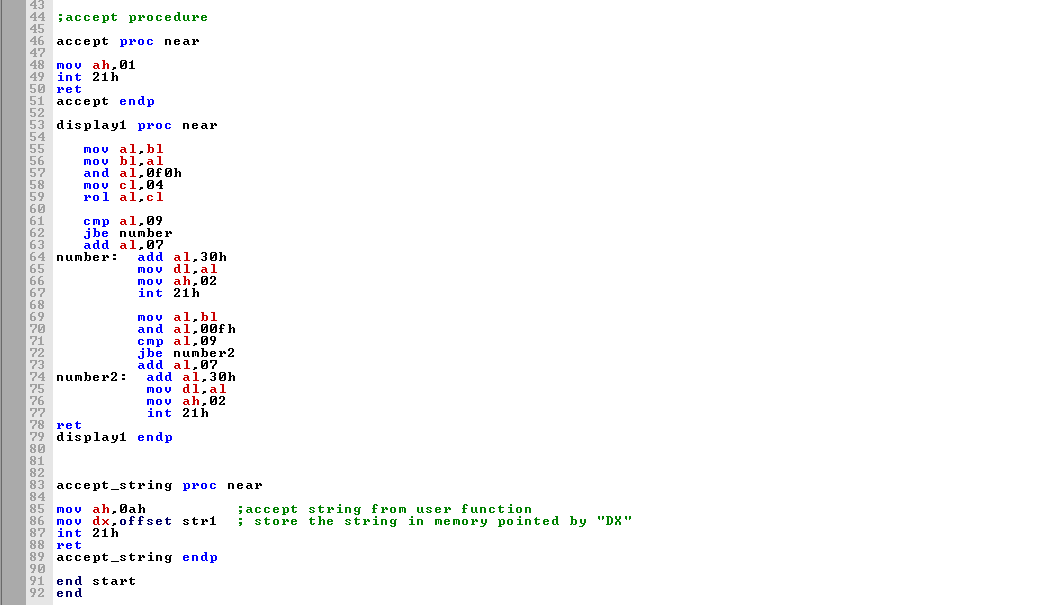
**int 21h**

**ret**

**accept\_string endp**

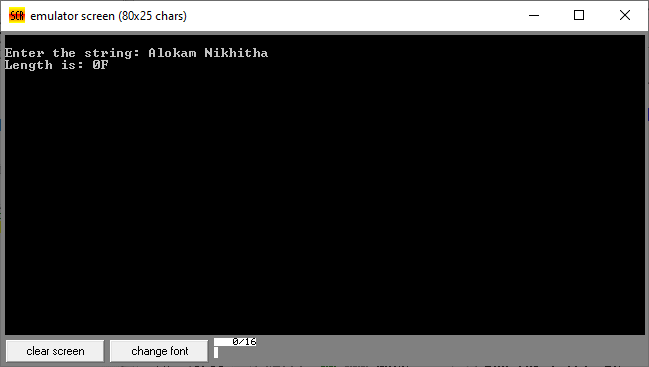
**end start**

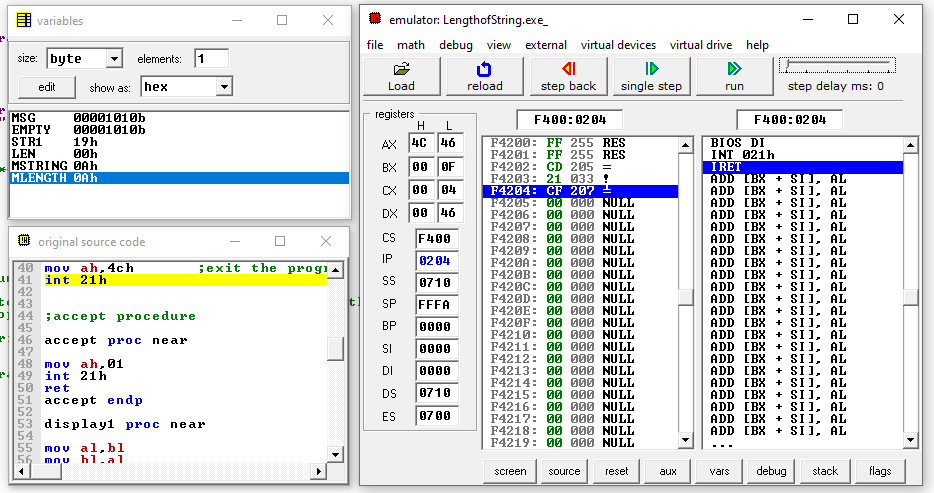
**end**

**** ****

**Variables before Execution**

**output**

****

****

**Result in BX**

b) count of spaces in string

**CODE**

**;macro for printing a string**

**print macro m**

**mov ah,09h**

**mov dx,offset m**

**int 21h**

**endm**

**.model small**

**;\*\*\*\*\*\* Data Segment \*\*\*\*\*\***

**.data**

**empty db 10,13, " $"**

**str1 db 25,?,25 dup('$')**

**mstring db 10,13, "Enter the string: $"**

**mscount db 10,13, "Number of spaces: $"**

**mlength db 10,13, "Length is: $"**

**scount db ?**

**;\*\*\*\*\*\*\*\*\*\* Code Segment \*\*\*\*\*\*\*\*\*\*\*\***

**.code**

**start:**

**mov ax,@data**

**mov ds,ax**

**print mstring**

**call accept\_string**

**mov si,offset str1+2 ;position si to start of the string**

**mov cl,str1+1 ;copy length in cl**

**mov dh,00 ;counter to store number of spaces**

**cmpagain1: mov al,[si] ;copy content at memory location "si" in "al"**

**cmp al,' ' ;compare "al" with space**

**jne below ;if not equal jump to label "below"**

**inc dh**

**below: inc si ;move to next character**

**dec cl ;decrement string length counter**

**jnz cmpagain1 ;if not zero check again**

**mov scount,dh ;save the count in memory location "scount"**

**mov bl,scount ;copy count to "bl" for printing**

**print mscount**

**call display1**

**exit:**

**mov ah,4ch ;exit the program**

**int 21h**

**;accept procedure**

**accept proc near**

**mov ah,01**

**int 21h**

**ret**

**accept endp**

**display1 proc near**

**mov al,bl**

**mov bl,al**

**and al,0f0h**

**mov cl,04**

**rol al,cl**

**cmp al,09**

**jbe number**

**add al,07**

**number: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**mov al,bl**

**and al,00fh**

**cmp al,09**

**jbe number2**

**add al,07**

**number2: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**ret**

**display1 endp**

**accept\_string proc near**

**mov ah,0ah ;accept string from user function**

**mov dx,offset str1 ; store the string in memory pointed by "DX"**

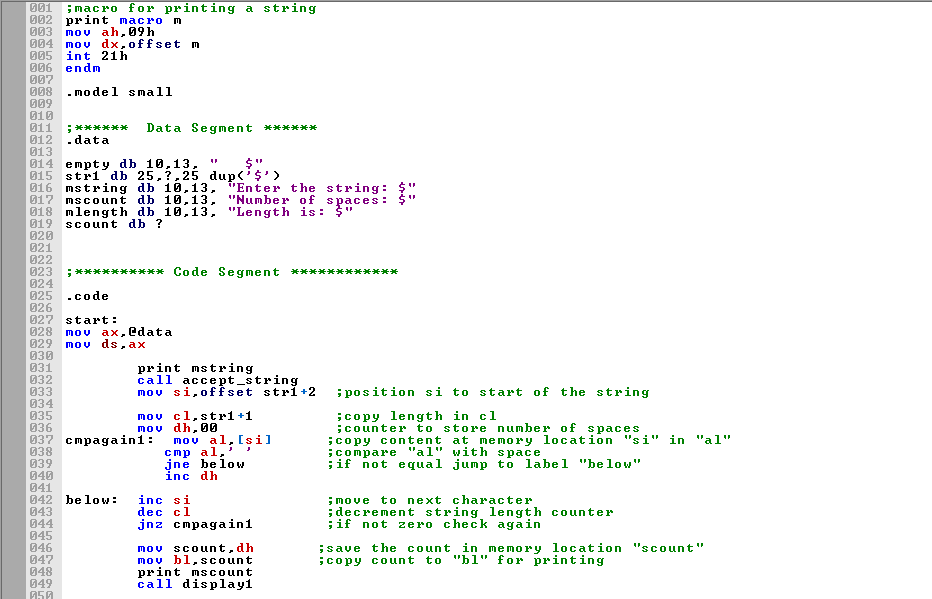
**int 21h**

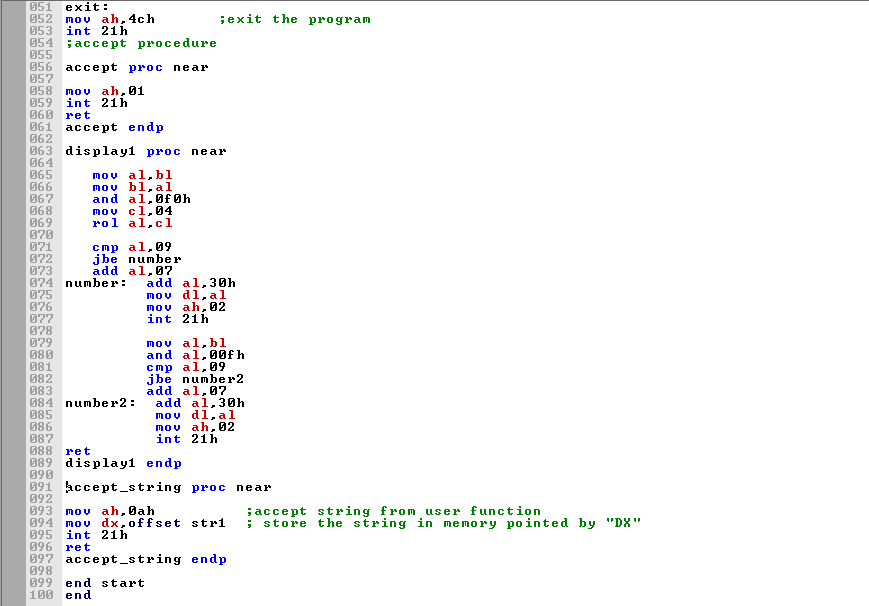
**ret**

**accept\_string endp**

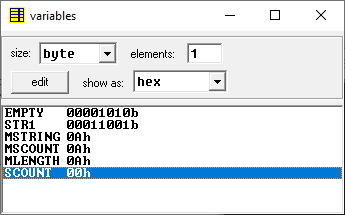
**end start**

**end**

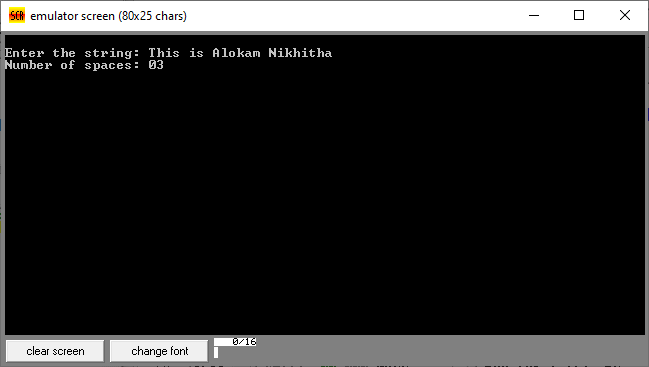
****

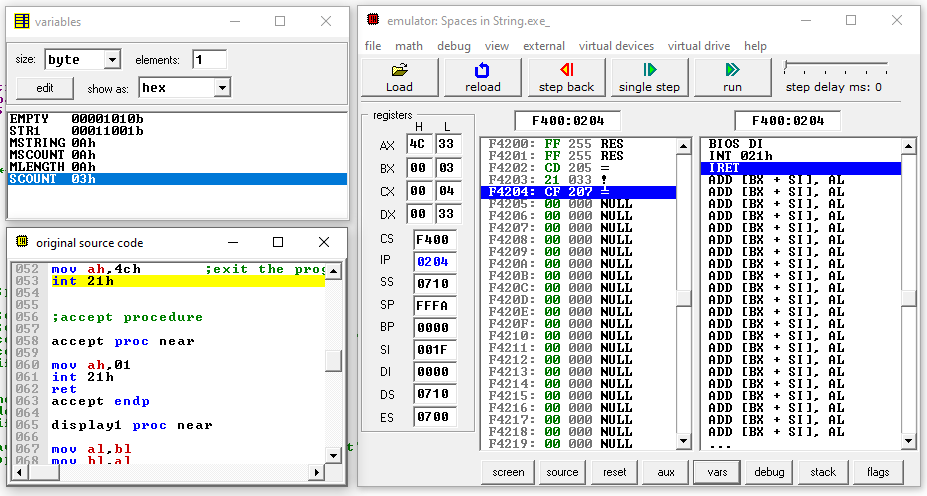
****

**Variables before Execution**

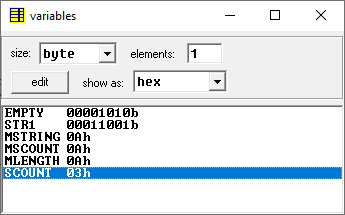
****

**Output**

****

****

**Variables after Execution**



**Result in Scount**

c) Reverse string and print it

**;macro for printing a string**

**print macro m**

**mov ah,09h**

**mov dx,offset m**

**int 21h**

**endm**

**.model small**

**;\*\*\*\*\*\* Data Segment \*\*\*\*\*\***

**.data**

**empty db 10,13, " $"**

**str1 db 25,?,25 dup('$')**

**str2 db 25,?,25 dup('$')**

**mstring db 10,13, "Enter the string: $"**

**mreverse db 10,13, "Reversed string: $"**

**;\*\*\*\*\*\*\*\*\*\* Code Segment \*\*\*\*\*\*\*\*\*\*\*\***

**.code**

**start:**

**mov ax,@data**

**mov ds,ax**

**print mstring**

**call accept\_string**

**mov si,offset str1 ;point si to start of string1**

**mov di,offset str2 ;point di to start of string2**

**mov al,[si] ;copy first two locations of string1 to string2**

**mov [di],al ;since these contain the size and length of the string**

**inc si ;which are same in reverse string also**

**inc di**

**mov al,[si]**

**mov [di],al**

**inc si**

**inc di**

**mov cl,str1+1 ; copy length in cl**

**mov ch,00**

**add si,cx ;add length of string1 to si to move it to last location**

**dec si ;si at last location of string1**

**move\_more: mov al,[si] ;copying character one by one from string1 pointed by si**

**mov [di],al ; to string2 pointed by "di" in reverse order as si moves**

**dec si ; from last character to first character**

**inc di**

**dec cl**

**jnz move\_more**

**print mreverse**

**print str2+2 ; printing the reversed string**

**print empty**

**exit:**

**mov ah,4ch ;exit the program**

**int 21h**

**;accept procedure**

**accept proc near**

**mov ah,01**

**int 21h**

**ret**

**accept endp**

**display1 proc near**

**mov al,bl**

**mov bl,al**

**and al,0f0h**

**mov cl,04**

**rol al,cl**

**cmp al,09**

**jbe number**

**add al,07**

**number: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**mov al,bl**

**and al,00fh**

**cmp al,09**

**jbe number2**

**add al,07**

**number2: add al,30h**

**mov dl,al**

**mov ah,02**

**int 21h**

**ret**

**display1 endp**

**accept\_string proc near**

**mov ah,0ah ;accept string from user function**

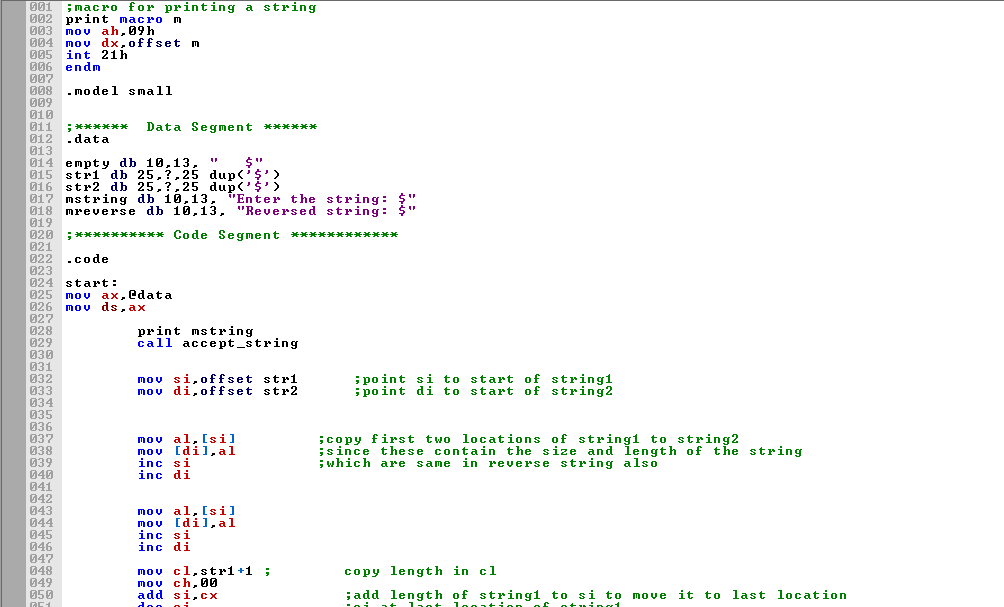
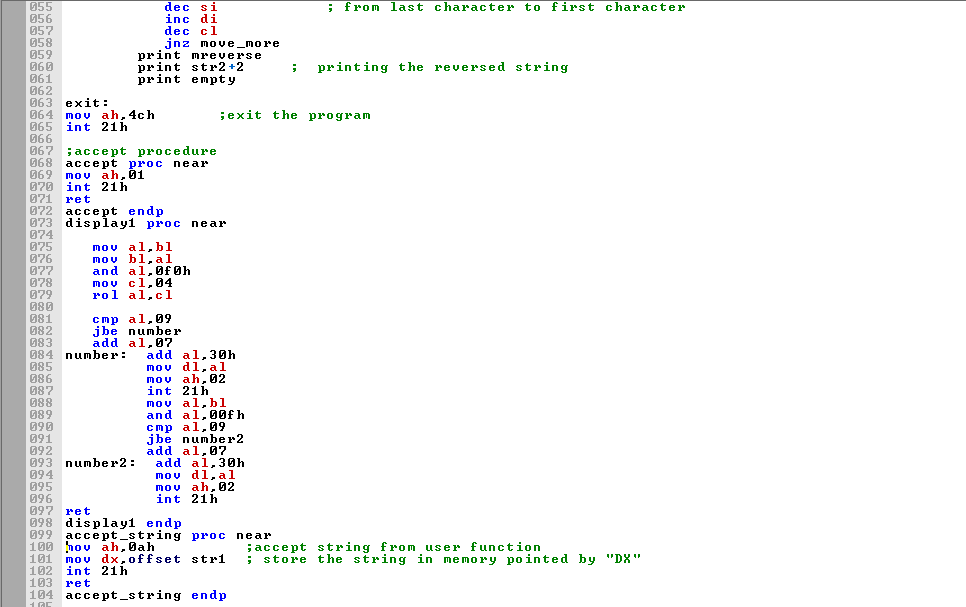
**mov dx,offset str1 ; store the string in memory pointed by "DX"**

**int 21h**

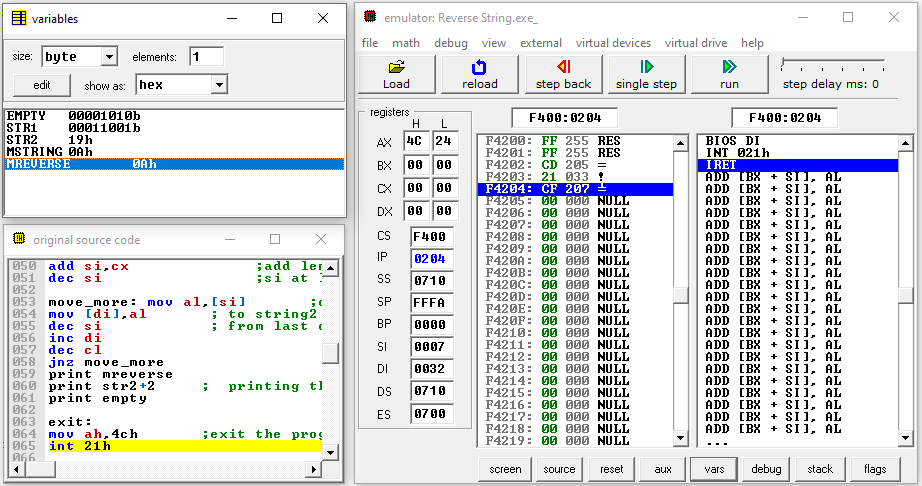
**ret**

**accept\_string endp**

**end start**

**end** **** 

**Output**

****

****