



Computer Organization & Assembly Language

LAB 13

Submitted to:

Ma'am Rehana

Submitted by:

Mahnoor Zahid

66572

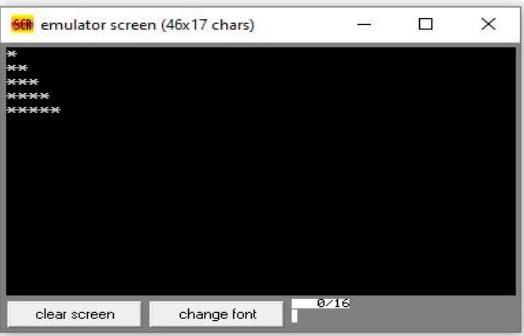
BSCS-3

Task 1

emu8086 - assembler and microprocessor emulator 4.08

```
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
```

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03 ; Mahnoor Zahid
04 org 100h
05 .model small
06 .stack 100h
07 .data
08 .code
09 main proc
10     mov ax, @data
11     mov ds, ax
12     mov cx, 5
13     mov bx, 1
14
15 loop:
16     push cx
17     call Stars
18     call newline
19     inc bx
20     pop cx
21     loop loop
22
23     mov ah, 4Ch
24     int 21h
25
26 main endp
27 Stars proc
28 Starsloop:
29     mov dl, '*'
30     mov ah, 2
31     int 21h
32     loop Starsloop
33     ret
34 Stars endp
35 newline proc
36     mov dl, 10
37     mov ah, 2
38     int 21h
39     ret
40 newline endp
41 end main
```



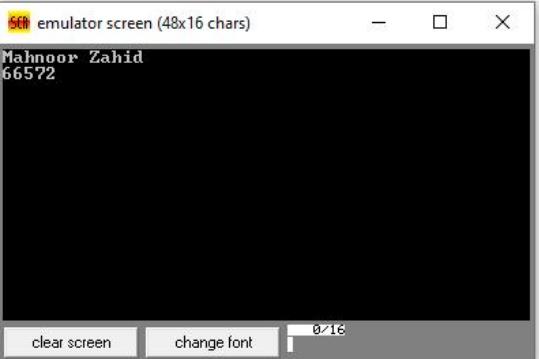
The screenshot shows the assembly code for Task 1. The code initializes the stack at 100h, sets up the data segment, and enters a loop. Inside the loop, it pushes the current value of CX onto the stack, calls the 'Stars' procedure, and then calls the 'newline' procedure. The 'Stars' procedure prints a single asterisk (*) using AH=02h and DL=48h (ASCII 'A'). The 'newline' procedure prints a carriage return (CR) and a line feed (LF) using AH=09h and DL=0Dh and DL=0Ah respectively. The assembly code ends with an INT 21h instruction to exit the program.

Task 2

emu8086 - assembler and microprocessor emulator 4.08

```
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about
```

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03 ; Mahnoor Zahid
04 string macro p1
05     mov dx, offset p1
06     mov ah, 9
07     int 21h
08     string endm
09
10
11 .data
12 stri db 'Mahnoor Zahid$'
13 str2 db '66572$'
14
15 .code
16 main proc
17     mov ax, @data
18     mov ds, ax
19
20     string stri
21     call print
22
23     string str2
24     call print
25
26     main endp
27 print proc
28     mov dl, 10
29     mov ah, 2
30     int 21h
31
32     mov dl, 13
33     mov ah, 2
34     int 21h
35
36     ret
37     print endp
38     end main
39
40 ret
```



The screenshot shows the assembly code for Task 2. The code defines two strings: 'stri' containing 'Mahnoor Zahid\$' and 'str2' containing '66572\$'. It then enters the 'main' procedure where it prints both strings using the 'print' procedure. The 'print' procedure uses AH=09h to output the string to the screen. The assembly code ends with an INT 21h instruction to exit the program.

Task 3

emu8086 - assembler and microprocessor emulator 4.08

file edit bookmarks assembler emulator math ascii codes help

new open examples save compile emulate calculator converter options help about

```
01 ; You may customize this and other start-up templates;
02 ; The location of this template is c:\emu8086\inc\0_com_template.txt
03 ; Mahnoor Zahid
04 .model small
05 .stack 100h
06
07 .data
08 msg1 db 'string 1',13,10,'$'
09 msg2 db 'string 2',13,10,'$'
10 msg3 db 'string 3',13,10,'$'
11 msg4 db 'string 4',13,10,'$'
12 msg5 db 'string 5',13,10,'$'
13
14 list dw offset msg1, offset msg2, offset msg3, offset msg4, offset msg5
15
16 .code
17
18 main proc
19     mov ax, @data
20     mov ds, ax
21
22     mov cx, 5
23     mov si, offset list
24
25 print_loop:
26     mov dx, [si]
27     call print_msg
28     add si, 2
29     loop print_loop
30
31     mov ah, 4ch
32     int 21h
33 main endp
34
35 print_msg proc
36     mov ah, 09h
37     int 21h
38     ret
39 print_msg endp
40
41 end main
42
```

emulator screen (50x18 chars)

string 1
string 2
string 3
string 4
string 5

clear screen change font 0/16