DEPARTMENT OF COMPUTER &
SOFTWARE ENGINEERING

COLLEGE OF E&ME, NUST, RAWALPINDI

Microprocessor and Microcontroller Based Design

Lab 01

SUBMITTED TO:
Dr Taimoor Zahid

SUBMITTED BY:
Muawiz Umer
Reg # 0000335806
DE-42 (C&SE)-A

Submission Date: 24/10/2022

Objectives:

In this lab, you will be introduced to memory segmentation, and difference between physical and logical addresses of the memory. You will also deal with the different types of addressing modes and learn how to calculate the physical addresses from logical addresses. You will also write programs that will make use of loops and jump statements.

Related Topic/Chapter in theory class:

None

Hardware/Software required:

Hardware: PC Software Tool: emu8086 v2.57

Tasks:

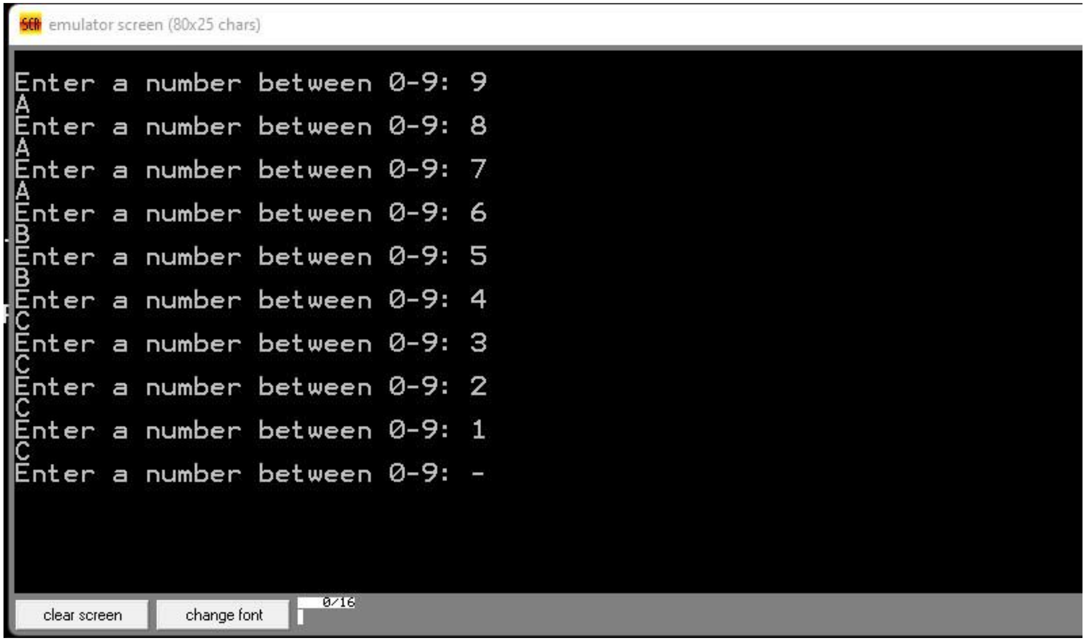
1. Write a program that takes a number input between 0 and 9 and then displays corresponding grade If grade is less than 5 it should display Grade C If grade is less than 7 it should display Grade B If grade is greater than or equal 7 it should display Grade A. The program should continue to run until user enters a negative number or a number greater than 9. Attach screenshot of all cases.

Solution:

```
org 100H
.STACK 100H


.DATA

   PROMPT  DB  13, 0ah, 'Enter a number between 0-9: $'
   print1  DB  13, 0ah, 'A $' ,13, 0ah
print2  DB  13, 0ah, 'B $' ,13, 0ah     print3
DB  13, 0ah, 'C $' ,13, 0ah


.CODE

loop1:

   MOV AX, @DATA              ; initialize DS
   MOV DS, AX

   LEA DX,PROMPT               ;ask uder prompt
   MOV AH,9
   INT 21H


   mov ah,1
   int  21h

   mov CL,AL      ;user input
sub CL, 30H
```

```
   cmp cl , 0
   jl exit

   cmp cl, 10
   jg exit

   mov bl,5
   cmp CL, BL       ;initialize bh=7
   jl C

   mov BL, 7
cmp BL, CL     jle
AGRADE
   cmp CL, BL
   jl B
exit:
   mov ah,4ch
int 21h
                          ;Result : ZF and CF set to ==> "ZF = 1" and "CF = 0"
                          ;Result : ZF and CF set to ==> "ZF = 0" and "CF = 0"
AGRADE:
    lea dx, print1              ;A if BL > BH =7
    mov   ah,9
int          21h
jmp loop1
                           ;Result : ZF and CF set to ==> "ZF = 0" and "CF = 1"
    B:
    lea dx, print2              ;B if BL < BH =7
    mov   ah,9
int          21h
jmp loop1
    C:

    lea dx, print3              ;C if  BL < BH =5
    mov ah,9
int 21h
    jmp loop1


ret
```

Output:



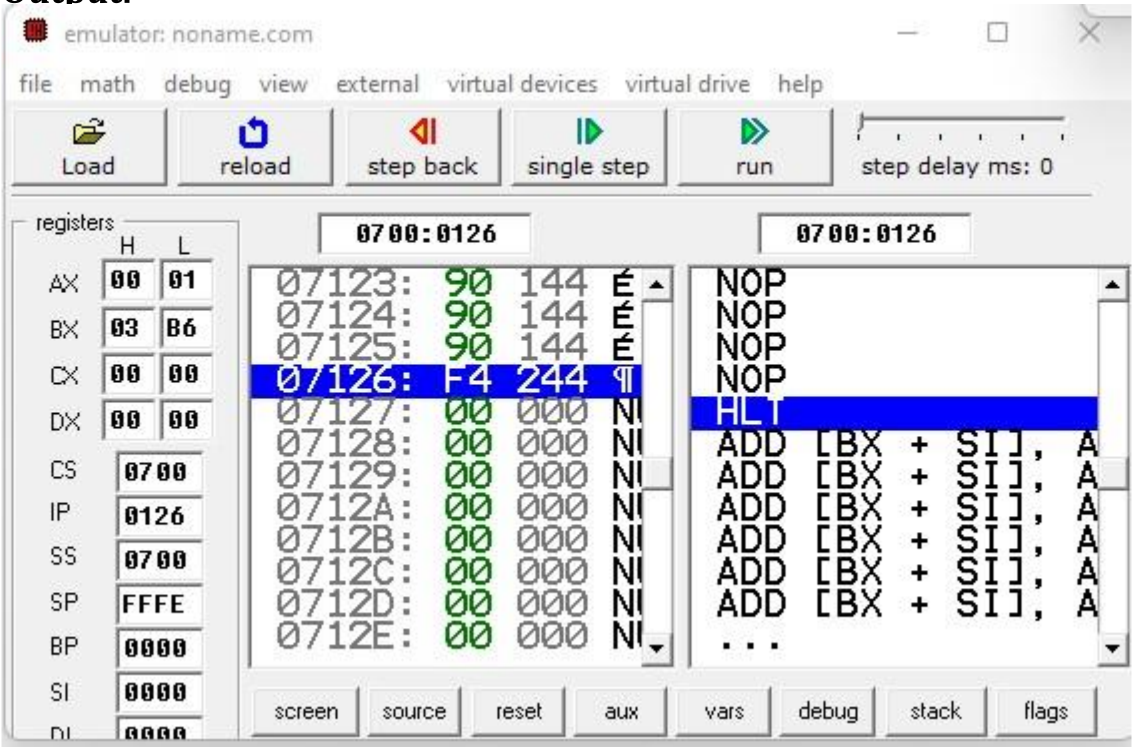
```
Enter a number between 0-9: 9
A
Enter a number between 0-9: 8
A
Enter a number between 0-9: 7
A
Enter a number between 0-9: 6
B
Enter a number between 0-9: 5
B
Enter a number between 0-9: 4
C
Enter a number between 0-9: 3
C
Enter a number between 0-9: 2
C
Enter a number between 0-9: 1
C
Enter a number between 0-9: -
```

2. **Write a program that adds following series and places result in AX. => 95+90+85+…+5**

Solution:

```
3.  org 100h
4.  .data
5.  .code
6.  main proc
7.
8.
9.     mov  ax, 1
10.    mov cx,95
11.    mov bx, 0    ;This puts zero in AX
12.    Label1:
13.    add bx, cx
14.    sub cx, 4           ; This adds int turn 1, 2, 3, ... ,10 to DX 15.     loop    Label1    16.
17.    end mainp
```

Output:



```
registers
     H    L
AX   00   01
BX   03   B6
CX   00   00
DX   00   00
CS   0700
IP   0126
SS   0700
SP   FFFE
BP   0000
SI   0000
DI   0000
```

```
0700:0126                    0700:0126
07123:  90  144  É           NOP
07124:  90  144  É           NOP
07125:  90  144  É           NOP
07126:  F4  244  ¶           NOP
07127:  00  000  NU          HLT
07128:  00  000  NU          ADD  [BX + SI], A
07129:  00  000  NU          ADD  [BX + SI], A
0712A:  00  000  NU          ADD  [BX + SI], A
0712B:  00  000  NU          ADD  [BX + SI], A
0712C:  00  000  NU          ADD  [BX + SI], A
0712D:  00  000  NU          ADD  [BX + SI], A
0712E:  00  000  NU          ...
```

Conclusion:

The LOOP instruction is a combination of a DEC and JNZ instructions. It causes execution to branch to the address associated with the LOOP instruction.