Pundra University of Science & Technology

CSE 328: Microprocessor Systems and Interfacing Lab

Course Teacher: **Md. Ataur Rahman**

Lecturer, Dept. of CSE, PUB

---

1. **Write a program to display any input to the output by using assembly language and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.CODE

 MAIN PROC

   MOV AH,1
   INT 21H          ;INPUT
   MOV BL,AL

   MOV AH,2
   MOV DL,BL     ;OUTPUT
   INT 21H

  MOV AH, 4CH
  INT 21H          ;EXIT
  MAIN ENDP
 END MAIN
```

2. **Write a program by using assembly language to display any input to the output by inserting newline and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.CODE

 MAIN PROC

   MOV AH,1
   INT 21H            ;INPUT
   MOV BL,AL

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

   MOV AH,2
   MOV DL,BL       ;OUTPUT
   INT 21H
```

```
  MOV AH, 4CH
  INT 21H              ;EXIT
  MAIN ENDP
END MAIN
```

3. **Write a program by using assembly language to display multiple input to the output and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

    MOV AH,1
    INT 21H

    MOV BL,AL        ;INPUT-1
    INT 21H

    MOV BH,AL        ;INPUT-2
    INT 21H

    MOV CL,AL        ;INPUT-3
    INT 21H

    MOV CH,AL       ;INPUT-4

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

    MOV DL, BL       ;OUTPUT-1
    INT 21H

    MOV DL, BH       ;OUTPUT-2
    INT 21H

    MOV DL, CL        ;OUTPUT-3
    INT 21H

    MOV DL, CH       ;OUTPUT-4
    INT 21H

    MOV AH, 4CH
    INT 21H
    MAIN ENDP
END MAIN
```

4. **Write a program by using assembly language to display any string and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'Pundra University Of Science & Technology $'
MSG2 DB 'Rangpur Road, Gokul, Bogura-5800 $'
.CODE

MAIN PROC

    MOV AX,@DATA
    MOV DS, AX

    LEA DX, MSG1
    MOV AH,9
    INT 21H

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

    LEA DX, MSG2
    MOV AH,9
    INT 21H

    MOV AH, 4CH
    INT 21H
    MAIN ENDP
END MAIN
```

5. **Write a program by using assembly language to add two numbers and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'ENTER VALUE1: $'
MSG2 DB 'ENTER VALUE2: $'
MSG3 DB 'SUM OF TWO NUMBERS ARE: $'
.CODE

MAIN PROC

   MOV AX,@DATA
   MOV DS, AX

   LEA DX, MSG1
   MOV AH,9
   INT 21H
```

```
MOV AH,1
INT 21H
MOV BL,AL

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

LEA DX, MSG2
MOV AH,9
INT 21H

MOV AH,1
INT 21H
MOV CL,AL

ADD BL,CL

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

LEA DX, MSG3
MOV AH,9
INT 21H

 MOV AH,2
 SUB BL, 48

 MOV DL,BL
 INT 21H


 MOV AH, 4CH
 INT 21H
 MAIN ENDP
END MAIN
```

6. **Write a program by using assembly language to subtract two numbers and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'ENTER VALUE1: $'
MSG2 DB 'ENTER VALUE2: $'
MSG3 DB 'SUBTRACTION OF  $'
MSG4 DB ' FROM $'
MSG5 DB ' IS: $'
.CODE
```

```
MAIN PROC

    MOV AX,@DATA
    MOV DS, AX

    LEA DX, MSG1
    MOV AH,9            ;ENTER VALUE1:
    INT 21H

    MOV AH,1
    INT 21H             ;1ST INPUT TO BL
    MOV BL,AL

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

    LEA DX, MSG2        ;ENTER VALUE2:
    MOV AH,9
    INT 21H

    MOV AH,1
    INT 21H             ;2ND INPUT TO CL
    MOV CL,AL

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

    LEA DX, MSG3
    MOV AH,9            ;SUBTRACTION OF
    INT 21H

    MOV AH,2            ;2ND VALUE
    MOV DL,CL
    INT 21H

    LEA DX, MSG4        ; FROM
    MOV AH,9
    INT 21H

    MOV AH,2            ;1ST VALUE
    MOV DL,BL
    INT 21H

    LEA DX, MSG5        ;IS
    MOV AH,9
    INT 21H

    SUB BL,CL           ;BL = SUB
```

```
    MOV AH,2
    ADD BL, 48

    MOV DL,BL
    INT 21H

    MOV AH, 4CH
    INT 21H
     MAIN ENDP
END MAIN
```

7. **Write a program by using assembly language for variable declaration and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
VAR1 DB 5
VAR2 DB ?              ; " ? " MEANS THAT WE CAN INPUT ANY VARIABLE
.CODE

MAIN PROC

    MOV AX,@DATA
    MOV DS, AX

    MOV AH,1
    INT 21H              ;INPUT VARIABLE FROM KEYBOARD
    MOV VAR2,AL

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

    MOV AH,2
    ADD VAR1, 48      ;ADD 48 TO 5 (1ST VARIABLE) TO GET ALLUDED OUTPUT
    MOV DL,VAR1
    INT 21H

    MOV DL,VAR2
    INT 21H

    MOV AH, 4CH
    INT 21H
    MAIN ENDP
END MAIN
```

8. **Write a program by using assembly language for case conversion (Conversion of Character to value, i.e., A=10, B=11, C=12,….., J=19) and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
VAR DB ?
.CODE

MAIN PROC

   MOV AX,@DATA
   MOV DS, AX

   MOV AH, 1          ;INPUT VARIABLE FROM KEYBOARD LIKE A,B,C,etc.
   INT 21H
   MOV VAR, AL

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

   MOV AH, 2
   MOV DL, '1'           ;SHOWING THE FIXED NUMBER 1
   INT 21H

   MOV AH, 2
   SUB VAR, 17        ;SUB 65-17= 48 TO SHOW THE ALLUDED OUTPUT LIKE A=0, B=1….
   MOV DL, VAR
   INT 21H

   MOV AH, 4CH
   INT 21H
   MAIN ENDP
END MAIN
```

Assembly Language Key Words Introduction

CMP = Compare JG = Jump If Greater Than JL = Jump If Less JGE = Jump If Greater Than Or Equal JNG = Jump If Not Greater Than JE = Jump If Equal JNE = Jump If Not Equal JMP = Jump INC = Increase DEC = Decrease

9. **Write a program by using assembly language for compare 2 values (greater/smaller) and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'ENTER AN INPUT: $'
MSG2 DB '   GREATER $'
MSG3 DB '   SMALLER $'
.CODE

MAIN PROC

   MOV AX,@DATA
   MOV DS,AX

   LEA DX,MSG1
   MOV AH,9
   INT 21H

   MOV AH,1
   INT 21H
   MOV BL,AL
   SUB BL,48

   CMP BL,5
   JL L1

   LEA DX,MSG2
   MOV AH,9
   INT 21H

   JMP EXIT

   L1:

   LEA DX,MSG3
   MOV AH,9
   INT 21H

    EXIT:

   MOV AH,4CH
   INT 21H
   MAIN ENDP
END MAIN
```

10. **Write a program by using assembly language for matching the specific value when it terminate the input insertion from keyboard and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

TOP:
   MOV AH,1
   INT 21H              ;INPUT
   MOV BL,AL

   MOV AH,2
   MOV DL,BL          ;OUTPUT
   INT 21H

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

   CMP BL, 23H          ;MATCHING VALUE
   JE EXIT
   JMP TOP

EXIT:

   MOV AH,4CH
   INT 21H
   MAIN ENDP
END MAIN
```

11. **Write a program by using assembly language for testing the value "is the value in range?" and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.DATA
MSG1 DB 'ENTER AN INPUT: $'
MSG2 DB '  IN RANGE $'
MSG3 DB '  OUT OF RANGE $ '
.CODE

MAIN PROC

   MOV AX,@DATA
   MOV DS, AX

   LEA DX, MSG1
   MOV AH, 9
   INT 21H
```

```
MOV AH, 1
INT 21H
MOV BL, AL
SUB BL, 48       ;SHOWING THE OUTPUT VALUE IN DECIMAL

CMP BL, 4
JGE L2           ;Jump If Greater Than Or Equal

L1:

 LEA DX, MSG3
 MOV AH, 9
 INT 21H

 JMP EXIT

L2:

  CMP BL, 7
  JNG L3         ;Jump If  Not Greater Than
  JMP L1

L3:

 LEA DX,MSG2
 MOV AH,9
 INT 21H

EXIT:

 MOV AH,4CH
 INT 21H
 MAIN ENDP
END MAIN
```

12. Write a program by using assembly language for "print the greatest number among three inputs" and run this program in the emulator.

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

   MOV AH,1
   INT 21H

   MOV BL, AL       ;INPUT-1
   INT 21H

   MOV BH, AL       ;INPUT-2
   INT 21H

   MOV CL, AL       ;INPUT-3
```

```
    CMP BL, BH     ;Compare btn BL & BH, if BL ≥ BH then jump to level L2
    JGE L2         ;If BL ≥ BH then jump to level L2, otherwise jump to level L1

    L1:

    CMP BH, CL   ;Compare btn BH & CL
    JGE LL1      ;If BH ≥ CL then jump to level LL1

    MOV AH, 2
    MOV DL, CL   ;If CL ≥ BH then print the value of CL
    INT 21H

    JMP EXIT

   LL1:

    MOV AH, 2
    MOV DL, BH  ; Print the value of BH
    INT 21H

    JMP EXIT

   L2:

    CMP BL, CL
    JGE LL2        ;Compare btn BL & CL, if BL ≥ CL then jump to level LL2

    MOV AH, 2
    MOV DL, CL     ;If CL ≥ BL, then print the value of CL
    INT 21H

    JMP EXIT

   LL2:

    MOV AH, 2
    MOV DL, BL    ; Print the value of BL
    INT 21H

    JMP EXIT

  EXIT:

   MOV AH,4CH
   INT 21H
   MAIN ENDP
END MAIN
```

13. Write a program by using assembly language for "using loop and print the user defined stars" and run this program in the emulator.

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

   MOV AH,1
   INT 21H
   MOV BL, AL
   SUB AL, 30H

   MOV CX, 0      ;initialization for how many stars are printing, using 0 indicates the user defined value
   MOV CL, AL

   LOOP:

   MOV AH, 2
   MOV DL, '*'

  TOP:

   INT 21H
   LOOP TOP

  EXIT:

   MOV AH,4CH
   INT 21H
   MAIN ENDP
END MAIN
```

14. Write a program by using assembly language for "using FOR loop for descending order" and run this program in the emulator.

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

    MOV AH,1
    INT 21H
    MOV BL, AL
    SUB AL, 30H

    MOV CX, 0     ;It can be written in the form " XOR CX,CX  "
    MOV CL,AL
```

```
TOP:

    MOV AH,2
    MOV DL,BL
    INT 21H
    DEC BL
    LOOP TOP

  EXIT:

  MOV AH,4CH
  INT 21H
  MAIN ENDP
END MAIN
```

15. **Write a program by using assembly language for "ascending order print" and run this program in the emulator.**

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

  MOV AH,1
  INT 21H
  MOV BL,AL

  MOV CL,'1'

TOP:

  MOV AH,2
  MOV DL,CL
  INT 21H
  INC CL
  CMP BL,CL
  JE EXIT
  JMP TOP

EXIT:

  MOV AH,4CH
  INT 21H
  MAIN ENDP
END MAIN
```

16. Write a program by using assembly language for "using NESTED LOOP and print the stars in multiline" and run this program in the emulator.

```
.MODEL SMALL
.STACK 100H
.DATA
.CODE

MAIN PROC

    MOV CX,4        ;LINE NUMBER
    MOV BX,2        ;NUMBER OF STAR

    TOP:

    MOV AH,2
    MOV DL,'*'
    INT 21H
    DEC BX

    CMP BX,0
    JE EXIT

    JMP TOP

  EXIT:

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

  MOV BX,2
  LOOP TOP

  MOV AH,4CH
  INT 21H
  MAIN ENDP
END MAIN
```

17. Write a program by using assembly language for "print ascii table using NESTED FOR loop" and run this program in the emulator.

```
.MODEL SMALL
.STACK 100H
.CODE

MAIN PROC

   MOV CX,0

   L1:
   MOV BX,CX
   MOV CX,8

   L2:
   MOV AH,2
   MOV DL,BL
   INT 21H

   INC BL
   CMP BL,255
   JE EXIT
   LOOP L2

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

  INC BL
  MOV CX,BX

  LOOP L1

EXIT:

   MOV AH,4CH
   INT 21H
   MAIN ENDP
END MAIN
```