Name: _____ Roll #:_____ Section:_____

1. In case of Logical Shift instructions, the empty bits are filled with
   a. 0
   b. 1
   c. 2
   d. 3
2. In case of Arithmetic Shift Left instruction, the empty bits are filled with
   a. 0
   b. 1
   c. 2
   d. 3
3. Shifting any operand 2 bits towards left actually multiplies it by
   a. 1
   b. ½
   c. 4
   d. 8
4. SAR instruction fills the empty bits with
   a. LSB
   b. Sign-bit
   c. 1
   d. None of the above
5. How many bits should we rotate an 8-bit value to swap its nibbles?
   a. 2 bits left
   b. 2 bits right
   c. 4 bits left
   d. 6 bits left

Name: _____ Roll #:_____ Section:_____

Consult the following code to answer next four questions

```
1       .DATA
2           a DB 10H
3           aptr DW a
4       .CODE
5       MAIN PROC
6           MOV AL, a
7           MOV BL, [aptr]
8           MOV SI, aptr
9           MOV AL, [SI]
10          MOV DL, 10
11          MAIN ENDP
12      END MAIN
```

6. Which addressing mode is used to access variable `a` in line number 6?

    a. Register addressing mode

    b. Direct addressing mode

    c. Immediate addressing mode

    d. None of the above

7. Which addressing mode is used to access register `AL` in line number 6?

    a. Register addressing mode

    b. Indirect addressing mode

    c. Register indirect addressing mode

    d. None of the above

8. Which addressing mode is used in line number 9 to access the second operand?

    a. Register addressing mode

    b. Immediate addressing mode

    c. Register indirect addressing mode

    d. None of the above

9. Which addressing mode is used in line number 10 to access the second operand?

    a. Register addressing mode

    b. Immediate addressing mode

    c. Register indirect addressing mode

    d. None of the above

10. Direct access to which segment is required in direct addressing mode?

    a. Data segment

    b. Code segment

    c. Stack segment

    d. Both a and b

11. In indirect addressing mode, the address of operand is placed in

    a. IP

    b. Memory location

    c. PC

    d. None of the above

12. Minimum number of operands in x86 assembly language instruction is

    a. 0

    b. 1

    c. 2

    d. None of the above

13. CPU understands the function performed by any instruction by its

    a. Offset

    b. IP

    c. Operation Code

    d. All of the above

14. 2-dimentional array can be accessed using which two addressing modes?

    a. Immediate and Register

    b. Indirect and Direct

    c. Base Index and Base Index Displacement

    d. None of the above

Name: _____ Roll #:_____ Section:_____

15. Which two registers are used to point to indexes of string by default

    a. AX and BX

    b. SI and DI

    c. CS and SI

    d. DI and AX

16. Which are two basic design issues in instruction format?

    a. Operation Code and Instruction Length

    b. Instruction Length and Allocation of Bits

    c. Allocation of Bits and Operation Code

    d. None of the above

17. What is the value of `r_size`

```
.data
    a DW 10h, 20h, 30h, 40h
    r_size = ($ - a)
```

    a. 2

    b. 4

    c. 6

    d. 8

18. Which flag are affected in Shift/Rotate instructions?

    a. PF and SF

    b. CF and OF

    c. CF and ZF

    d. All of the above

Congratulation, you got two free points ☺

Name: _____ Roll #:_____ Section:_____

Q1.

(2+2+2+2)

You are given the following x86 assembly language program. What will be the value of registers AX, BX, CX and DX after the program execution completes?

```
ORG 100H
.MODEL SMALL
.STACK 100H
.DATA
    a DB 26h, 27h, 28h, 29h, 30h
.CODE
MAIN PROC

    MOV CX, 2
    MOV SI, OFFSET a
    MOV AL, [SI]
    MOV AH, [SI]
    CALL EX_PROC
    MOV DL, [SI]
    MOV DH, DL

    RET
    MAIN ENDP

EX_PROC PROC

    PUSH AX
    CLD
    REP MOVSW
    POP BX

    RET
    EX_PROC ENDP

END MAIN
```

```
AX = 2626h

BX = 2626h

CX = 0h

DX = 3030h
```

Name: _____ Roll #:_____ Section:_____

Q2.

```
1   .CODE
2   MAIN PROC
3       MOV CX, 1
4       MOV AL, 00001111b
5       RCR AL, 1
6       RCL AL, 1
7       RCR AL, 1
8       JZ LABEL_1
9       JS LABEL_2
10      JC LABEL_3
11      MOV BL, 00001111b
12      LABEL_1:
13          OR AL, 00000001b
14          LOOP LABEL_1
15      LABEL_2:
16          MOV DL, 00111100b
17          STC
18          JZ LABEL_6
19          JC LABEL_5
20          MOV BL, 99
21          JMP LABEL_EXIT
22      LABEL_3:
23          RET
24          JMP LABEL_EXIT
25      LABEL_4:
26          MOV BL, 0
27          JZ LABEL_2
28          JP LABEL_6
29          JMP LABEL_EXIT
30      LABEL_5:
31          XOR AL, 00001000b
32          JC LABEL_5
33          JP LABEL_4
34          MOV CL, 3
35          JMP LABEL_EXIT
36      LABEL_6:
37          CMP AL, 128
38          JE LABEL_4
39          JA LABEL_1
40          JS LABEL_3
41          MOV BL, 11110000b
42          JMP LABEL_EXIT
43      LABEL_EXIT:
44          .EXIT
45  MAIN ENDP
```

Name: _____ Roll #:_____ Section:_____

```
46  END MAIN
```

a. You are given a program above. Read this code and write the contents in any one format (decimal/hex/binary) of the following registers when the program execution completes. (2+2+2+2)

```
AL = 07h
BL = 00h
CL = 01h
DL = 00h
```

b. Line numbers are given against each instruction in the above code. Trace the order of execution of the program and write down line numbers in which the program will execute. (2+8)

3, 4, 5, 6, 7, 8, 9, 10

22, 23

```
Line number written in
red colour is optional.
```

Name: _____ Roll #:_____ Section:_____

Q3. What will be contents of AL and AH when following x86 assembly language instructions are executed? Answer should be in decimal. (4)

```
MOV AX, 36765
MOV DX, 287
MOV BL, 100
DIV BL
```

The actual answer is overflow error because the quotient cannot fit inside 8-bits AL register.

But if a student calculates the quotient and remainder and writes them in the respective registers, he/she will be awarded points for this answer.

```
AL = 367
AH = 65
```