1. Suppose stack segment is delared: .STACK 100h

a. Hex contents of SP when program begins?

```
 SP = 0100
```

b. What is max hex number of words that stack may contain?

```
100h bytes / 2_bytes_per_word = 80h words
```

2. Suppose that AX=1234h ,BX=5678h, CX=9ABCh, and SP=0100h . Give the contents of AX,BX, CX, and SP after executing the following instructions:

```
    PUSH AX
    PUSH BX
    XCHG AX,CX
    POP  CX
    PUSH AX
    POP  BX
```

```
Initial:   AX=1234h BX=5678h CX=9ABCh SP=0100h
```

	AX	BX	CX	SP
PUSH AX	1234	5678h	9ABCh	00FE
PUSH BX	1234	5678h	9ABCh	00FC
XCHG AX,CX	9ABCh	5678h	9ABCh	00FC
POP CX	9ABCh	5678h	5678h	00FE
PUSH AX	9ABCh	5678h	5678h	00FC
POP BX	9ABCh	9ABCh	5678h	00FE

3- Write some code to do the following:

a- Place the top of the stack into AX without changing the stack contents

```
contents
POP  AX     ;place top of stack into AX
PUSH AX     ;restore stack contents
```

b- Place the word that is below the stack top into CX, without changing the stack contents. You may use AX.

```
POP  AX     ;place original stack top into AX
POP  CX     ;place word that is below original stack top into CX
PUSH CX     ;restore word that is below original stack top
PUSH AX     ;restore original stack top
```

c- Exchange the top two words on the stack. You may use AX and BX.

```
c. Exchange the top two words on the stack. You may use AX and BX.
POP  AX     ;place original stack top into AX
POP  BX     ;place word that is below original stack top into BX
PUSH AX     ;place original stack top back on stack
PUSH BX     ;place word that was originally below stack top onto stack
top
```

writ program to REVERSE INPUT from the user use push and pop instruction -4

```
    .MODEL  SMALL
    .386
    .STACK
    .CODE
    MAIN    PROC
    ;display user prompt
            MOV     AH,2            ;prepare to display
            MOV     DL,'?'          ;char to display
            INT     21H             ;display '?'
    ;initialize character count
            XOR     CX,CX           ;count = 0
    ;read a character
            MOV     AH,1            ;prepare to read
            INT     21H             ;read a char
    ;while character is not a carriage return do
    WHILE_:
            CMP     AL,0DH          ;CR?
            JE      END_WHILE       ;yes, exit loop
    ;save character on the stack and increment count
            PUSH    AX              ;push it on stack
            INC     CX              ;count = count + 1
    ;read a character
            INT     21H             ;read a char
            JMP     WHILE_          ;loop back
    END_WHILE:
    ;go to a new line
            MOV     AH,2            ;display char fcn
            MOV     DL,0DH          ;CR
            INT     21H             ;execute
            MOV     DL,0AH          ;LF
            INT     21H             ;execute
            JCXZ    EXIT            ;exit if no characters read
    ;for count times do
    TOP:
    ;pop a character from the stack
            POP     DX              ;get a char from stack
    ;display it
            INT     21H             ;display it
            LOOP    TOP
    ;end_for
    EXIT:
            MOV     AH,4CH
            INT     21H
    MAIN    ENDP
```

```
             END      MAIN
```

5- Write a procedure for finding the product of two positive integers A and B by addition and bit shifting.

Multiplication algorithm:

```
Product = 0
Repeat
If 1sb of B is 1 (Recall 1sb = least significant bit)
Then
Product = Product + A
End_if
Shift left A
Shift right B
UNTIL B = 0
```

```
        .MODEL  SMALL
        .386
        .STACK
        .CODE
        MAIN    PROC
        ;execute in 386DBG. Place A in AX and B in BX.
             CALL       MULTIPLY
        ;DX will contain the product
             MOV        AH,4CH
             INT        21H
        MAIN    ENDP
        MULTIPLY        PROC
        ;multiplies two nos. A and B by shifting and addition
        ;input: AX = A, BX = B. Nos. in range 0 - FFh.
        ;output: DX = product
                PUSH    AX
                PUSH    BX
                XOR     DX,DX           ;product = 0
        REPEAT_:
        ;if B is odd
                TEST    BX,1            ;is B odd?
                JZ      END_IF          ;no, even
        ;then
                ADD     DX,AX           ;prod = prod + A
        END_IF:
                SHL     AX,1            ;Shift left A
                SHR     BX,1            ;Shift right B
        ;until
                JNZ     REPEAT_
                POP     BX
                POP     AX
                RET
        MULTIPLY       ENDP
```

```
         END      MAIN
```

6- suppose DX contains 000h , AX contains 0005h and BX contains 0002h

```
DIV BX

AX = 0002

DX =0001

IDIV BX

AX = 0002

DX = 0001
```

7- divide the signed value of the byte variable XBYTE by -7

```
MOV AL , XBYTE

CBW

MOV BL , -7

IDIV BL
```