#P1_1:Write an assembly language program to find sum of 10 random numbers#

```
.MODEL SMALL
.STACK 64
.DATA

SZ   DB 10                          ;size of array
ARR  DB 1,2,12,15,14,5,6,7,8,9      ;array elements
SUM    DB ?                                  ;difference to be stored

      .CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            LEA SI,ARR      ;load Memory Location of First element into SI
            LEA DI,SZ
            MOV CL,[DI]       ;Initialize CX with size of array (Here,10)
            MOV AL,00         ;Initialize AL to zero

LOOP1:      ADD AL,[SI]       ;Add data into accumulator
            INC SI            ;Increment pointer to array element
            DEC CL                  ;Decrement counter
            JNZ LOOP1         ;Loop again if CX is non zero
            MOV SUM,AL      ;Store the answer in SUM
            HLT
            MOV AH,4CH
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point
```

OUTPUT : SUM = 4FH

#P1_2: Write an assembly language program to find average of the given set of 16 bit number numbers#

```
.MODEL SMALL
.STACK 64
.DATA

SZ   DW 8                         ;size of array
ARR  DW 12,15,14,5,6,7,8,9      ;array elements
AVG    DW ?                              ;difference to be stored

      .CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            MOV CX,SZ         ;store size in CX as a counter
            MOV BX,CX       ;store size for division
            LEA SI,ARR      ;load address of start of array to SI
            MOV AX,0000       ;initialze AX with zero

LOOP1:      ADD AX,[SI]       ;Add current element to AX
```

```
            INC SI                  ;Increment SI
            INC SI
            DEC CX                  ;Decrement Counter CX
            JNZ LOOP1          ;if CX is non zero go to LOOP

            DIV BX                  ;Divide AX with BX to get average
            MOV AVG,AX         ;Move it to AVG
            HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN           ;this is the program exit point
```

OUTPUT : AVG = 09H

#P1_3:Write a program to find largest and smallest among an array of numbers and print the difference between them#

```
.MODEL SMALL
.STACK 64
.DATA

SZ   DB 8                        ;size of array
ARR  DB 12,15,14,5,6,7,8,9       ;array elements
DIFF DB ?                                ;difference to be stored

      .CODE
MAIN  PROC  FAR          ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            MOV CL,SZ         ;store size in CL
            LEA SI,ARR      ;load address of start of array to SI
            MOV AL,[SI]       ;Move First element to AL
            MOV BL,[SI]       ;Move First element to AL

MIN :   CMP AL,[SI]           ;Compare AL with current element
            JC MAX                  ;if AL is smaller go to MAX
            MOV AL,[SI]       ;else mov that element to AL

MAX : CMP BL,[SI]       ;Compare BL with current element
            JNC LOOP1         ;if BL is larger go to LOOP1
            MOV BL,[SI]       ;elese mov that element to BL

LOOP1 : INC SI                ;Increment SI to position of next element in the
array
            DEC CX                  ;Decrement counter(CX)
            JNZ MIN           ;If CX is non zero go to MIN
            SUB BL,AL         ;subtract smallest from largest
            MOV DIFF,BL       ;move it to DIFF
            HLT

            MOV   AH,4CH            ;set up to
```

```
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point
```

OUTPUT : DIFF = 0AH

#P1_4a:Program to count number of 1's in a byte#

```
.MODEL SMALL
.STACK 64
.DATA

NUM  DB 28       ;Given number
CNT    DB ?        ;count to be stored

      .CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            SUB BL,BL         ;clear BL to keep number of 1s
            MOV DL,8          ;counter to rotate total 8 times
            MOV AL,NUM      ;store NUM value in AL

LOOP1 : ROL AL,1        ;rotate it once
            JNC GO          ;if Carry Flag = 0, go to GO
            INC BL                  ;if Carry Flag = 1, add one to count
GO :  DEC DL                ;Decrease counter
            JNZ LOOP1         ;if counter(DL) is non zero loop again
            MOV CNT,BL        ;Move count of 1's to CNT
            HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point
```

OUTPUT : CNT = 03H

#P1_4b:Program to reverse given array at same location#

```
.MODEL SMALL
.STACK 64
.DATA

ARR  DB 12,15,14,5,6,7,8,9      ;array elements

      .CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            MOV CX,0004       ;store half the size in CX
            LEA SI,ARR      ;load address of start of array to SI
```

```
            MOV DI,SI
            ADD DI,0007       ;Store memory location of last element in DI (Just
ADD size of array - 1 to SI)

LOOP1:   MOV AL,[SI]          ;Move current element to AL
            XCHG AL,[DI]      ;swap element at [DL] and in AL
            MOV [SI],AL       ;Move swapped element to [SI]
            INC SI                  ;Increment SI
            DEC DI                  ;Decrement DI
            DEC CX                  ;Decrement counter
            JNZ LOOP1
            HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point

OUTPUT : 09H, 08H, 07H, 06H, 05H, EH, FH, CH


#P1_4c:Program to find Factorial of a number#

.MODEL SMALL
.STACK 64
.DATA

NUM DW 0005H

.CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
          LEA SI,NUM          ;load address of num into SI
          LEA DI,NUM+100H     ;load memory location NUM+100H into DI
          MOV CX,[SI];  ;Move value at SI, given number to CX
            MOV AX,0001;      ;Initialize AX with 1
            MOV DX,0000;      ;Initialize DX with 0

LOOP1 :     MUL CX                  ;Multiply AX with current value of CX
            DEC CX                  ;Decrement CX
            JNZ LOOP1         ;If CX is non zero loop again

            MOV [DI],AX;      ;Move value in AX to memory location stored in DI
            MOV [DI+1],DX;    ;Move value in DX to DI+1
            HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point

OUTPUT : 78H (Factorial of 0005H)


#P1_4d:Program To find the no of even & odd nos. from given array of nos.#

.MODEL SMALL
.STACK 64
```

```
.DATA

SZ       DB 8                        ;size of array
ARR      DB 12,13,14,5,6,7,8,9       ;array elements
CNTODD       DB ?                              ;cnt of odd numbers
CNTEVEN      DB ?                              ;cnt of even numbers

.CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
          LEA SI,ARR          ;load address of num into SI
            MOV BX,0000       ;Initialize BX with 0
            LEA DI,SZ
            MOV CL,[DI]       ;Initialize CX with size of array

LOOP1:  MOV AL,[SI]     ;Load data into Accumulator
        AND AL,01H      ;AND with 01H
        JZ  GO          ;If AND is 0, Jump to EVEN
        INC BL          ;Increment BL (stores cnt of odd)
        JMP NEXT        ;Jump to next
GO    :   INC BH          ;Increment BH (stores snt of even)
NEXT:   INC SI          ;Increment SI
        DEC CL               ;Decrement CL
        JNZ LOOP1       ;Loop until CL is non zero
            HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point

OUTPUT : CNTODD = 04H, CNTEVEN = 04H


#P1_4e:Program To check for a Palindrome (single letter)#

.MODEL SMALL
.STACK 64
.DATA

SZ       DB 7                                ;size of array
ARR      DB 'A','B','C','D','C','B','A'     ;array elements

.CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV AX,@DATA    ;Load Data in temp register
        MOV DS,AX       ;Load data into data into Data Segment
        LEA SI,ARR      ;Load address of first element into SI
        LEA DI,ARR+06H  ;Load address of last element into DI (i.e ARR + SZ-1)
        MOV CL,03H      ;Load half the size of ARR into CL
        MOV CH,00H           ;Initialize CH to 0

LOOP1:  MOV AH,[SI]     ;Load data into AH
        MOV BH,[DI]     ;Load data into BH
        CMP AH,BH       ;Compare AH and BH
        JNZ GO          ;If not zero skip
        INC SI          ;Increment pointer
        DEC DI          ;Decrement pointer
```

```
        DEC CL          ;Decrement pointer
        JNZ LOOP1       ;If not zero, jump to back
        INC CH          ;Increment CH
GO:     HLT

            MOV   AH,4CH            ;set up to
            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point

; In this if finally, CH = 01H then given array is a palindrome

OUTPUT : CH = 01H;

#P1_4e:Addition of two 16-bit nos#

.MODEL SMALL
.STACK 64
.DATA

NUM1  DW 8514H
NUM2  DW 5362H
SUM   DW ?
CARRY DB 00H

      .CODE
MAIN  PROC  FAR         ;this is the program entry point
            MOV   AX,@DATA    ;load the data segment address
            MOV   DS,AX       ;assign value to DS
            MOV AX,NUM1     ;Move from NUM1 into accumulator
          ADD AX,NUM2     ;Add NUM2 to AX
          JNC SKIP        ;If there is no carry, skip
          INC CARRY       ;else Increment carry

SKIP:   MOV SUM, AX     ;Store the answer
            HLT

            INT   21H               ;return to DOS
MAIN        ENDP
            END MAIN          ;this is the program exit point


OUTPUT : SUM = D876H and CARRY = 00H
```