8-bit Arithmatic Operations 24/12/19

Aim :- To perform arithmatic operations between two 8-bit numbers using MASM software.

Apparatus :- A system with MASM and 8086 assembler

program :-

```
ASSUME   DS: DATA , CS: CODE
DATA   SEGMENT
 NUM1    DB    16H    5678H
 NUM2    DB    07H
 R1      DB    ?
 R2      DB    ?
 R3      DB    ?
 R4      DB    ?
 DATA   ENDS
 CODE   SEGMENT
 START:  MOV   AX, DATA
         MOV   DS, AX
         MOV   AL, NUM1
         MOV   BL, NUM2
         ADD   AL, BL
         MOV   R1, AL
         MOV   AL, NUM1
         SUB   AL, BL
```

```
MOV     R2, AL
MOV     AL, NUM1
MUL     AL, BL
MOV     R3, AL
MOV     AL, NUM1
DIV     AL, BL
MOV     R4, AL
INT     03H
CODE    ENDS
END     START
```

Result:-

The arithmetic operations on two 8-bit numbers has been performed

24/12/19

16-bit Arithmatic Operations 24/12/19

Aim :- TO perform arithmatic operation between two 16-bit numbers using MASM software.

Appartus :- A system with MASM and 8086 assembler

program:-

```
ASSUME DS: DATA, CS: CODE
DATA SEGMENT
N1    DBW   5678H
N2    DBW   1234H
R1    DW     ?
R2    DW     ?
R3    DW     ?
R4    DW     ?
DATA  ENDS
CODE  SEGMENT
START :  MOV  AX, DATA
         MOV  DS, AX
         MOV  BL, NUM2
         ADD  AL, BL
         MOV  R1, AL
         MOV  AL, N1
         SUB  AL, BL
         MOV  R2, AL
         MOV  AL, N1
         MUL  AL, BL
```

```
MOV     R3, AL
DIV     AL, BL
MOV     R4, AL
INT     03H
CODE    ENDS
END     START
```

Result:-

The arithmetic operations on 16-bit numbers has been performed.

24/11/19

Ascending and Descending Order 31/10/17

Aim:- To perform or arrange given numbers in incrementing or decrementing order.

Apparatus:- A system with MASM software

program:-

```
ASSUME DS:DATA, CS:CODE
DATA SEGMENT
LIST . DB AOH, 1OH, F9H, 36H, 99H
DATA ENDS
CODE SEGMENT
START: MOV . AX, DATA
       MOV DS, AX .
       MOV CH, 04H
       ~~MOV SI, LIST~~
UP2 :  MOV CL, 04H
       LEA SI, LIST
UP1 :  MOV AL, [SI]
       MOV BL, [SI+1]
       CMP AL, BL
       JC  DOWN
       MOV DL, [SI+1]
       XCHG ~~AL, BL,~~ [SI], DL
       MOV [SI+1], DL
DOWN
  →    INC SI
       DEC CL
       JNZ UP1
       DEC CH
       JNZ UP2
       INT 03H
CODE ENDS
END START
```

16-bit Ascending and Desending Order

31/12/19

Aim: To perform or arrange given numbers in incrementing or decrementing order of 16-bit.

Apparatus:- A system with MASM software.

program:-

```
ASSUME  CS: CODE , DS: DATA
DATA    SEGMENT
LIST    DW    ABCDH, 1102H, 1234H, F001H, 9999H
DATA    ENDS
CODE    SEGMENT
START :   MOV    AX, DATA
          MOV    DS, AX
UP2 :     MOV    CH, 04H
          LEA    SI, LIST
UP1 :     MOV    SI , LIST
          MOV    CL, 04H
          MOV    AX, [SI]
          MOV    BX, [SI +2]
          EMP    AX, BX
          JC     DOWN
          MOV    DX, [SI+2]
          XCHG   [SI], DX
          MOV    [SI+2], DX
DOWN :    INC    SI
          INC    SI
          DEC    CL
          JNZ    UP1
          DEC    CH
          JNZ    UP2
          INT    03H
CODE      ENDS
END       START
```

Result:-

Program 3 :- (descending order)

```
ASSUME    DS: DATA , CS:CODE
DATA      SEGMENT
LIST      DB       A0H, 10H,F9H, 36H ,99H
DATA      ENDS
CODE      SEGMENT
START :   MOV      AX, DATA
          MOV      DS, AX
          MOV      CH, 04H
          MOV      SI , LIST
UP2 :     MOV      CL ,04H
          LEA      SI , LIST
UP1 :     MOV      AL, [SI]
          MOV      BL, [SI +1]
          CMP      AL ,BL
          JNC      DOWN
          MOV      DL, [SI+1]
          XCHG     [SI], DL
          MOV      [SI+1],DL
DOWN →    INC      SI
          DEC      CL
          JNZ      UP1
          DEC      CH
          JNZ      UP2
          INT      03H
          CODE     ENDS
          END      START
```

program 4 :- (descending 16 bit)

```
ASSUME     CS : CODE , DS: DATA
DATA       SEGMENT
LIST       DW    ABCDH , 1102H , 1234H, F0dH , 9999H
DATA       ENDS.
CODE       SEGMENT
START :        MOV    AX, DATA
               MOV    DS, AX
 UP2 :     MOV     SI, LIST
           LEA     SI , LIST
           MOV     SI , LIST
 UP1 :     MOV     CL, 04H
           MOV     AX, [SF]
           MOV     BX , [SI+2]
           EMP      AX, BX
           JNC      DOWN
           MOV      DX, [SI+2]
           XCHG     [SI], DX
           MOV      [SI+2], DX
 DOWN :     INC     SI
            INC     SI
            DEC     CL
            JNZ     UPI .
            DEC     CH
            JINZ    UP2
            INT      03H
            CODE     ENDS
             END     START
```

Result :- The descending and ascending on 16-bit has been performed.

5/12/24

Searching a number in an array   7/01/2020

Aim:- find the given number is available in the array or not and print the message.

Apparatus:- PC with MASM software.

Program:-

```
ASSUME   CS:CODE , DS: DATA
DATA   SEGMENT
LIST    DB      10H,20H 30H, 40H, 50H, 60H,70H
SRC    DB   40H
MSG1   DB → "Number found$"
MSG2   DB → "Number Not found $"
DATA   ENDS.
CODE   SEGMENT
START :  MOV   AX,DATA
         MOV  DS, AX
         LEA  SI , LIST
         MOV   CL, 07H
UP :  MOV   BL, [SI]
      MOV  AL, SRC
      CMP   AL,BL
      JZ , GOT
      INC   SI
      DEC   CL
      JNZ   UP
      PRINT    MSG2
      JMP    END
GOT :   PRINT  MSG1
END :  INT    03H
       CODE   ENDS
       END    START

PRINT  MACRO  MSG
MOV    AH, 09H
```

```
LEA   DX, MSG
INT   21H
INT   03H
ENDM
```

Result: Searching a number in an array has been performed.

7/1/2020

4. Logical operations:

07/01/2020

Aim:- To perform logical operations btw two 8-bit numbers.

Apparatus:- A system with MASM and 8086 assembler.

Program:-

```
ASSUME CS:CODE , DS:DATA
DATA SEGMENT
N1   DB  1234H
N2   DB  2024H
R1   DB  ?
R2   DB  ?
R3   DB  ?
DATA ENDS
CODE SEGMENT
START : MOV AX, DATA
        MOV DS, AX
        MOV AL, N1
        MOV BL, N2
        AND AL, BL
        MOV R1, AL
        MOV AL, N1
        OR  AL, BL
        MOV R2, AL
        MOV AL, NUM1
        NOT AL
        MOV R3, AL
        MOV AL, NUM1
        XOR AL, BL
        MOV R4, AL
        INT 03H
        CODE ENDS
        END START
```

Result:- Logical operations btw 2 numbers have been performed

5. String Manipulation Instructions

28/01/2020

Aim:- Right hand ALP to perform the reverse of a string

Apparatus:- A PC with MASM software

program:-

```
ASSUME  CS:CODE, DS:DATA
DATA  SEGMENT
MSG   db  'string Reverse $'
CNT   dw    $-MSG
REV   db    ?
DATA  ENDS
CODE  SEGMENT
START :  MOV   AX,DATA
         MOV   DS, AX
         MOV   ES, AX
         MOV   CX, CNT
         DEC   CX
         DEC   CX
         LEA   SI,MSG
         LEA   DI,REV
         LEA   SI, CNT
         DEC   SI
         DEC   SI
Repeat :  MOV   AL, [SI]
          Mov   [DI],AL
          DEC   SI
          INC   DI
          LOOP  REPEAT
          MOV   AL, [SI]
          Mov   [DI], AL
```

```
        INC   DI
        MOV   DL, '$'

PRINT : MOV   AH, 09H
        LEA   DX, REV
        INT   21H

EXIT  : MOV   AH, 4CH
        INT   21H
        CODE  ENDS
        END   START.
```

Result:- String Manipulations have been peformed.

28/1/2020

b) Aim: check given string is palindrome or not and print the message if given string is palindrome if not print the message given string is not palindrome.

Program:- ASSUME CS:CODE, DS:DATA, ES:EXTRA.

```
DATA SEGMENT
STRING  DB  'MALAYALAM $'
MSG1    DB  "GIVEN STRING IS PALINDROME $"
MSG2    DB  "GIVEN STRING IS NOT PALINDROME$"
PAL     DB  00H
DATA  ENDS
PRINT   MACRO MSG
MOV     AH, 09H
LEA     DX , MSG
INT     21H
INT     03H
ENDM
EXTRA  SEGMENT
BLOCK    DB  9  DUP(?)
EXTRA  ~~SEGMEN~~ENDS.
CODE  SEGMENT
BEGIN:  MOV  AX, DATA
        MOV  DS, AX
        MOV  AX , EXTRA
        MOV  ES, AX
        LEA  SI , STRING
        LEA  DI , BLOCK+8
        MOV  CX , 0009 H
BACK:   CLD     // Auto increment (clear Directional flag)
```

```
        LODSB           (load string in buffer).
        STD             (storage)
        STOSB           (compare)
        LOOP BACK
        LEA   SI, STRING
        LEA   DI, BLOCK
        MOV   CX, 0009H
        CLD
        REPZ  CMPSB
        JNZ   SKIP
        PRINT MSG1
SKIP:   PRINT MSG2
        CODE  ENDS
        END   BEGIN
```

Result :- string manipulations have been performed.

28/1/2020