MICROPROCESSORS PRACTICAL

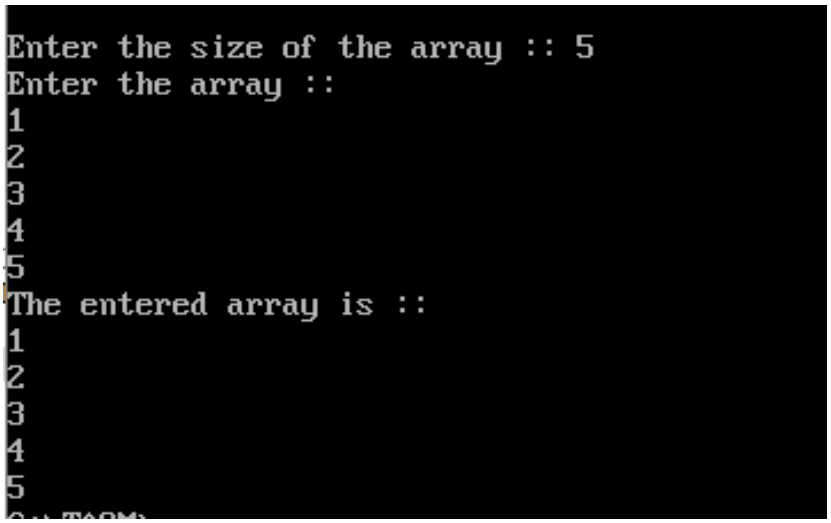
ARRAY PROGRAMS ::

1.Enter and print array of one digit.

```
.model small

.data
        ARRAY DB 10 DUP (?)
        DATA1 dw 0000H
        msg db 10,13,"Enter the size of the array :: $"
        msg2 db 10,13,"Enter the array :: $"
        msg3 db 10,13,"The entered array is :: $"

.code
.startup
        MOV AH,09
        MOV DX,OFFSET msg
        INT 21H

        MOV AH,01     ;input array size
        INT 21H
        SUB AL,30H
        MOV AH,0
        MOV CX,AX
        MOV DATA1,AX     ;store size in DATA1

        MOV AH,09
        MOV DX,OFFSET msg2   ;print msg2
        INT 21H

        MOV AH, 0
        MOV SI, 0
        LEA BX,ARRAY    ;store base address of the array in bx
        L1: MOV DL, 0AH ; jump onto next line
            MOV AH, 02H
            INT 21H

          MOV AH, 01H  ;input number
          INT 21H
          SUB AL,30H
          MOV AH, 00
          MOV [BX + SI], AX   ;store number in the array
          INC SI       ;incremenet si to point to the next memory location
        LOOP L1

        MOV AH, 09H
        LEA DX,MSG3    ;print msg3
        INT 21H

        MOV CX,DATA1   ;set cx = data1
        MOV SI, 0      ;set si to point at the base of the array
        L2: MOV AH,02h
          MOV DL,0AH   ;jump onto next line
          INT 21H

          MOV DL, 0dh
```

```
        INT 21H

        MOV DX,[BX+SI]
        ADD DL, 30h
        MOV AH, 02      ;print num
        INT 21H
        INC SI          ;increment si
      LOOP L2
.EXIT
END
```

OUTPUT

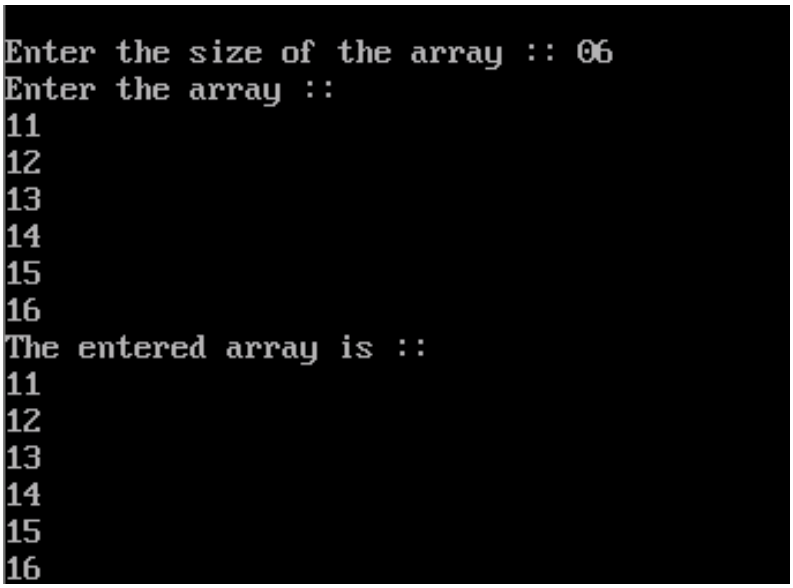


2. Enter and print array of two digits.

```
.model small

.data
        A1 Db 100  DUP (?)
        DATA1 dw 0000H
        DATA2 db 00h
        DATA3 db 00h
        msg db 10,13,"Enter the size of the array :: $"
        msg2 db 10,13,"Enter the array :: $"
        msg3 db 10,13,"The entered array is :: $"

.code
.startup
        MOV AH,09
        LEA DX,msg ;print msg
        INT 21H

        MOV CX, 2
        AGAIN: MOV AH,01   ;input size
          INT 21H
          CMP AL,'A'
       JGE L1
          SUB AL,30H
          JMP L2
          L1: SUB AL,37H
          L2: SHL BL, 4
          ADD BL, AL
        LOOP AGAIN

        MOV AL,BL
        MOV AH,00
        MOV DATA1, AX  ;store size in data1


        MOV AH,09
        MOV DX,OFFSET msg2  ;print msg2
        INT 21H

        LEA SI,A1
        MOV CX, DATA1
        L3: MOV DL, 0AH ; jump onto next line
          MOV AH, 02H
          INT 21H

          MOV DATA2,CL

          NUM PROC    ;create procedure
          MOV CL, 2
          AGAIN2: MOV AH, 01H
                  INT 21H
                  SUB AL,30H
                  SHL BL,4
                  ADD BL,AL
          LOOP AGAIN2
```

```
        ENDP NUM

        MOV CL, DATA2

        MOV [SI], BL
        INC SI
      LOOP L3

      MOV AH, 09H
      LEA DX,MSG3  ;print msg
      INT 21H

      MOV CX, DATA1
      MOV SI, offset A1
      L4: MOV AH, 02h     ;display number
     MOV DL, 0ah
        INT 21h

        MOV DATA3, CL
     MOV CL, 2
     MOV BL, [SI]
     AGAIN3: ROL BL, 4
          MOV AL, BL
          AND AL, 0FH
          MOV DL, AL
          ADD DX, 30h
          MOV AH, 02
                INT 21H
        LOOP AGAIN3

        MOV CL, DATA3
        INC SI
      LOOP L4
.EXIT
END
```

OUTPUT



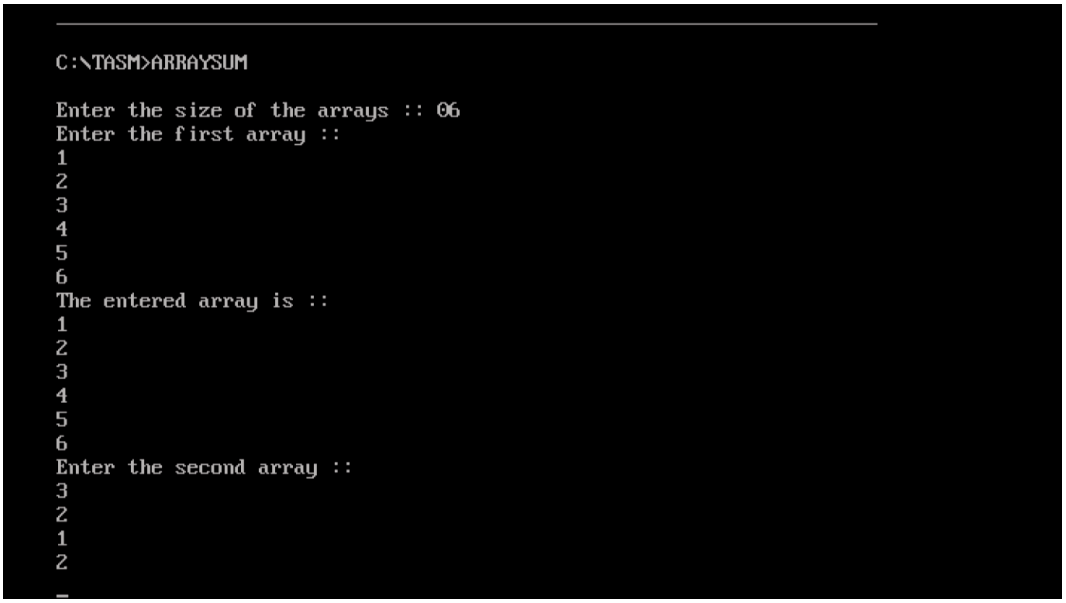
3. Add two arrays

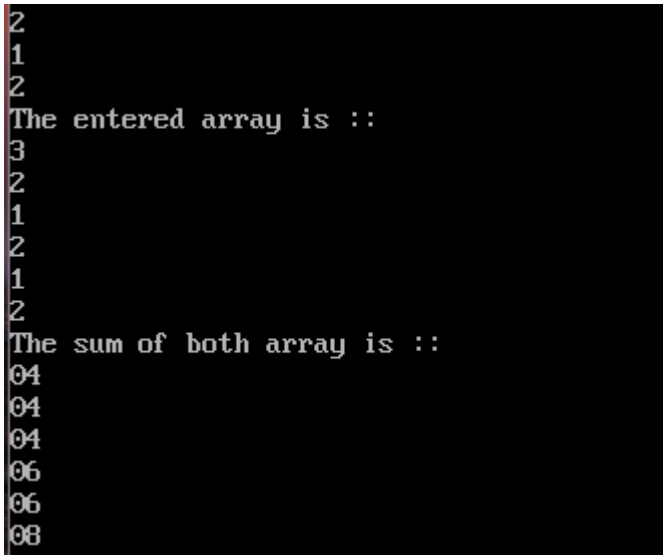
```
.model small
.386
.data
        A1 DB 20 DUP (?)
        A2 DB 20 DUP (?)
        DATA1 dw 0000H
        DATA2 DW 0000H
        msg db 10,13,"Enter the size of the arrays :: $"
        msg2 db 10,13,"Enter the first array :: $"
        msg3 db 10,13,"The entered array is :: $"
        msg4 db 10,13, "Enter the second array ::$"
        msg5 db 10,13, "The subtaction of both array is ::$"

.code
.startup
        MOV AH,09
        MOV DX,OFFSET msg
        INT 21H

        MOV CX, 2
        L4: MOV AH,01
           INT 21H
           CMP AL,'A'
           JGE L9
           SUB AL,30H
           JMP L8
           L9: SUB AL,37H
           L8: SHL BX, 4
           ADD BL, AL
           LOOP L4
           MOV AL, BL
           MOV CL, AL
           MOV AH, 0
           MOV DATA1, AX
           MOV CX, DATA1

        MOV AH,09
        MOV DX,OFFSET msg2
        INT 21H

        MOV CX, DATA1
        LEA SI, A1
        L1: MOV DL, 0AH ; jump onto next line
           MOV AH, 02H
           INT 21H

           MOV AH, 01H
           INT 21H
           SUB AL,30H
           MOV [SI], AL
           INC SI
        LOOP L1

        MOV AH, 09H
        MOV DX, OFFSET MSG3
```

```
INT 21H

MOV CX, DATA1
LEA SI, A1
L2: mov ah, 02h
   mov dl, 0ah
   INT 21h
   MOV DL, [SI]
   ADD DL, 30h
   MOV AH, 02
   INT 21H
   INC SI
LOOP L2

MOV CX, DATA1

MOV AH,09
MOV DX,OFFSET msg4
INT 21H
MOV AH,0

LEA DI, A2
L3: MOV DL, 0AH ; jump onto next line
   MOV AH, 02H
   INT 21H

   MOV AH, 01H
   INT 21H
   SUB AL,30H

   MOV [DI], AL
   INC DI
LOOP L3

MOV AH, 09H
MOV DX, OFFSET MSG3
INT 21H

MOV CX, DATA1
LEA DI, A2
L14: mov ah, 02h
    mov dl, 0ah
    INT 21h
    mov dl, 0dh
    INT 21h
    MOV DX, [DI]
    ADD DL, 30h
    MOV AH, 02
    INT 21H
    INC DI
LOOP L14


LEA SI, A1
LEA DI, A2

MOV CX, DATA1
SUBLOOP: MOV AL, [SI]
     SUB AL, [DI]
```

```
            MOV [SI], AL
            INC DI
            INC SI
        LOOP SUBLOOP

        MOV AH, 09H
        MOV DX, OFFSET MSG5
        INT 21H

        MOV CX, DATA1
        LEA SI, A1
        L5: mov ah, 02h
           mov dl, 0ah
           INT 21h
           MOV DATA2, CX
           MOV CX, 2
           MOV BL, [SI]

           SUBLOOP2: ROL BL, 4
                MOV DL, BL
                AND DL, 0FH
                CMP Dl, 9
                JA L6
                ADD DL, 30h
                JMP L7

                L6: ADD DL, 37H
                L7: MOV AH, 02
                     INT 21H
          LOOP SUBLOOP2
          MOV CX, DATA2
          INC SI
    LOOP L5
.EXIT
END
```

OUTPUT





4. Subtract two arrays

```
.model small
.386
.data
        A1 DB 20 DUP (?)
        A2 DB 20 DUP (?)
        DATA1 dw 0000H
        DATA2 DW 0000H
        msg db 10,13,"Enter the size of the arrays :: $"
        msg2 db 10,13,"Enter the first array :: $"
        msg3 db 10,13,"The entered array is :: $"
        msg4 db 10,13, "Enter the second array ::$"
        msg5 db 10,13, "The subtaction of both array is ::$"

.code
.startup
        MOV AH,09
        MOV DX,OFFSET msg
        INT 21H

        MOV CX, 2
        L4: MOV AH,01
           INT 21H
           CMP AL,'A'
           JGE L9
           SUB AL,30H
           JMP L8
           L9: SUB AL,37H
           L8: SHL BX, 4
           ADD BL, AL
           LOOP L4
           MOV AL, BL
           MOV CL, AL
           MOV AH, 0
```

```
   MOV DATA1, AX
   MOV CX, DATA1

MOV AH,09
MOV DX,OFFSET msg2
INT 21H

MOV CX, DATA1
LEA SI, A1
L1: MOV DL, 0AH ; jump onto next line
   MOV AH, 02H
   INT 21H

   MOV AH, 01H
   INT 21H
   SUB AL,30H
   MOV [SI], AL
   INC SI
LOOP L1

MOV AH, 09H
MOV DX, OFFSET MSG3
INT 21H

MOV CX, DATA1
LEA SI, A1
L2: mov ah, 02h
   mov dl, 0ah
   INT 21h
   MOV DL, [SI]
   ADD DL, 30h
   MOV AH, 02
   INT 21H
   INC SI
LOOP L2

MOV CX, DATA1

MOV AH,09
MOV DX,OFFSET msg4
INT 21H
MOV AH,0

LEA DI, A2
L3: MOV DL, 0AH ; jump onto next line
   MOV AH, 02H
   INT 21H

   MOV AH, 01H
   INT 21H
```

```
   SUB AL,30H

   MOV [DI], AL
   INC DI
LOOP L3

MOV AH, 09H
MOV DX, OFFSET MSG3
INT 21H

MOV CX, DATA1
LEA DI, A2
L14: mov ah, 02h
    mov dl, 0ah
    INT 21h
    mov dl, 0dh
    INT 21h
    MOV DX, [DI]
    ADD DL, 30h
    MOV AH, 02
    INT 21H
    INC DI
LOOP L14


LEA SI, A1
LEA DI, A2

MOV CX, DATA1
SUBLOOP: MOV AL, [SI]
     SUB AL, [DI]
     MOV [SI], AL
     INC DI
     INC SI
LOOP SUBLOOP

MOV AH, 09H
MOV DX, OFFSET MSG5
INT 21H

MOV CX, DATA1
LEA SI, A1
L5: mov ah, 02h
   mov dl, 0ah
 INT 21h
   MOV DATA2, CX
   MOV CX, 2
   MOV BL, [SI]

   SUBLOOP2: ROL BL, 4
```

```
            MOV DL, BL
            AND DL, 0FH
            CMP Dl, 9
            JA L6
            ADD DL, 30h
            JMP L7

              L6: ADD DL, 37H
            L7: MOV AH, 02
              INT 21H
        LOOP SUBLOOP2
          MOV CX, DATA2
          INC SI
         LOOP L5
.EXIT
END
```

OUTPUT