1. To find the largest number among array of number in an array

```
.model small
.stack 100h
.data
array db 45h,12h,5h,78h,12h
large db ?
.code
main proc
   mov ax,@data
   mov ds,ax
   mov bl,large
   mov cx,0005h
   lea si,array
   again:
   cmp bl,[si]
   jnc lll
   mov bl,[si]
   lll:    inc si
   loop again
   mov large,bl
   mov ax,4c00h
   int 21h
main endp
```

2. Add the list of 2 table and store in third table

```
 .model small
.stack 100h
.data
val1 db 44h,45h,48h,45h,75h
val2 db 44h,45h,48h,45h,75h
val3 db 5 dup(?)
.code
main proc
```

```
        mov ax,@data
        mov ds,ax
        mov bx,0000h
        ;mov cx,0005h
        repeat:
        mov al,val1[bx]
        add al,val2[bx]
        mov val3[bx],al
        inc bx
        loop repeat
        mov ax,4c00h
        int 21h
    main endp
```

3. To display string at the central of the screen

```
.model small
.stack 100h
.data
val db "wait for second..$"
.code
main proc
mov ax,@data
mov ds,ax
mov ah,02h
mov dh,0ch      ;Row mov dh,12
mov dl,27h              ;column mov dl,40
int 10h;Set the curser position
  mov ah,09h
lea dx,val
int 21h    ;display string pointed by dx
 mov ax,4c00h
 int 21h
 main endp
```

4. WAP in 8086 to sort 5 numbers in ascending order and descending order.(2062 baisakh).

```
 .model small
.stack 100h
.data
list db 10h,20h,4h,50h,01h
.code
main proc
  mov ax,@data
  mov ds,ax
  repeat:
  lea si,list
  mov bl,00h
  mov cx,0004h
  lll:
  mov al,[si]
  inc si
  cmp al,[si]
  jc nochange
  mov dl,[si]
  mov [si],al
  dec si
  mov [si],dl
  inc si
  mov bl,01h   ;keep flag =1
  nochange:
  loop lll
  dec bl
  jz repeat    ;yes do another pass
  mov ax,4c00h
 int 21h
  main endp
```

5. Taking input and displaying string

```
.model small
.stack 100h
.data
paralist LABEL BYTE
max db 20
act db ?
```

```
kbdat db 21 dup(' '),'$'
prompt db 'name is ','$'
.code
main proc
mov ax,@data
mov ds,ax

mov ah,0ah       ;for input to the string
lea dx,paralist
int 21h
mov ah,02h
mov dh,12
mov dl,40
int 10h
  mov ah,09h     ;display string
  lea dx,prompt
  int 21h

     mov ah,09h
     lea dx,kbdat
     int 21h
     mov ax,4c00h
     int 21h
 main endp
```

6. Display the string character wis

```
.model small
.stack 100h
.data
string db "kathmandu Engineering",'$'
.code
main proc
mov ax,@data
mov ds,ax
lea si,string
again:
mov dl,[si]  ;char to be displayed
cmp dl,24h  ;ascii value of $ is 24h i.e end of string
jz last
```

```
mov ah,02h  ;display character to output
int 21h
inc si
jmp again
last:
mov ax,4c00h
int 21h
main endp
```

7. WAP to display string with Background Blue and Foreground red is(note to display string with background and foreground we use character method)

```
.model small
.stack 100h
.data
string db "kathmandu Engineering",'$'
.code
main proc
mov ax,@data
mov ds,ax
lea si,string

mov dx,0000h
again:
mov ah,02h
int 10h
mov ah,09h ;display character with foreground and background color
mov al,[si]
cmp al,'$'
je last
mov bh,0
mov bl,14h  ;background and foreground
mov cx,01h
int 10h
inc si
inc dx
jmp again
last:
mov ax,4c00h
```

```
int 21h
```

8. WAP to read a string character wise without echo and display the character by converting the small case latter to uppercase

```
.model small
.stack 100h
.code
main proc
   mov ax,@data
   mov ds,ax
   a1:
   mov ah,08h
   inc al
   int 21h
   cmp al,0dh
   je a3
   cmp al,'a'
   jb a2
   cmp al,'z'
   ja a2
   sub al,20h
   a2:
   mov ah,2
   mov dl,al
   int 21h
   jmp a1
   a3:
   mov ah,4ch
   int 21h
   main endp
.data
```

9. .WAP to read string and convert the small case letter to upper case and display the converted string in next line.(2060 bhadra)

....

```
.model small
.stack 100h
.data
```

```
kbdat db 21 dup(' '),'$'
.code
main proc
mov ax,@data
mov ds,ax

mov ah,0ah ;string input to kbdat
lea dx,kbdat
int 21h

lea si,kbdat  ;pointing at starting of string

again:
mov al,[si]
cmp al,'$'     ;Comparing for last of string
je a3
 cmp al,0dh  ;Compare for enter
je a3
cmp al,'a'
jb a1
cmp al,'z'
ja a1
sub al,20h
mov [si],al
a1:
inc si
jmp again
a3:
mov ah,02h
mov dh,12
mov dl,40
int 10h


 mov ah,09h;Display string
 lea dx,kbdat
 int 21h
 mov ax,4c00h
 int 21h
 main endp
```

10.WAP to get an string input; count number of vowels and display msg "even vowels" on the screen if the count is even otherwise display "odd vowels"(2055 chaitra)

```
.model small
.stack 100h
.data
kbdat db 21 dup(' '),'$'
odd db "odd vowel",'$'
even db "even vowel",'$'
.code
main proc
mov ax,@data
mov ds,ax

mov ah,0ah
lea dx,kbdat
int 21h

lea si,kbdat
mov bl,00h

again:
mov al,[si]
cmp al,'$'     ;Compare for end of string
je a3
 cmp al,0dh  ;Compare for enter
je a3
cmp al,'a'
jb a1
cmp al,'z'
ja a1
sub al,20h    ;Convert to upper case
a1:
cmp al,'A'
jnz a4
inc bl
jmp a2
a4:
cmp al,'E'
```

```
jnz a5
inc bl
jmp a2
a5:
cmp al,'I'
jnz a6
inc bl
jmp a2
a6:
cmp al,'O'
jnz a7
inc bl
jmp a2
a7:
cmp al,'U'
jnz a2
inc bl
jmp a2
a2:
inc si
jmp again
a3:

mov ah,02h
mov dh,12
mov dl,40
int 10h
 rcr bl,01h
 jnc chan
 mov ah,09h
 lea dx,odd
 int 21h
 jmp l1
 chan:
 mov ah,09h
 lea dx,even
 int 21h
 l1:
 mov ax,4c00h
 int 21h
 main endp
```

11.Taking the string input and displaying each word of the string in separate line

```
.model small
.stack 100h
.data
string db 99 dup(?)
.code
main proc
   mov ax,@data
   mov ds,ax
   mov bl,00h
   lea si,string
   again:
   mov ah,01h
   int 21h
   cmp al,0dh  ;carrage return ie enter
   jz display
   mov [si],al
   inc si
   inc bl
   jmp again
      display:

   mov ax,0600h    ;For clearing screen
   mov bh,71h
   mov cx,0000h
   mov dx,184fh
   int 10h


   mov cl,bl
   mov ch,00h
   lea si,string
   repeat:
   mov dl,[si]
   cmp dl,20h  ;ascii value of space is 20h
   jz newline
```

```
    mov ah,02h
    int 21h
    inc si
    loop repeat
    jmp last
    newline:
    inc si
   mov dl,0ah                    ;Next line code
    mov ah,02h
    int 21h
    mov dl,0dh
    mov ah,02h
    int 21h
    dec cx
    jmp repeat
    last:    mov ax,4c00h
 int 21h
main endp
```

12..WAP in 8086 to read a string and separate the words from the string, display each word at the center of each line of a clear screen with blue back ground and cyan foreground.(2062 bhadra)

```
.model small
.stack 100h
.data
string db 15 dup(' '),'$'
.code
main proc
mov ax,@data
mov ds,ax

mov ah,0ah
lea dx,string
int 21h

mov ax,0600h    ;For clearing screen
mov bh,0
mov bl,17h
mov cx,0000h
mov dx,184fh
int 10h

   mov dx,0040h
   lea si,string

   again:
   mov ah,02h  ;for cursor position
   int 10h
   inc dl
   mov al,[si]
   cmp al,'$'
   je last
   cmp al,20h
   jnz chartodis
   inc dh      ;Next line curser positon
   mov dl,40h
   jmp next
   chartodis:
   mov ah,09h
   mov bh,0
   mov bl,13h         ;background and foreground
   mov cx,01h  ;repeating the number of al character
   int 10h
   next:
   inc si
   jmp again
      last:
 mov ax,4c00h
 int 21h
main endp
```

13.WAP in 8086 to read a single digit number and display the multiplication table of that number as 2 4 6 8 10 12 14 16 18 20 if the users enter digit 2.(2067 shrawan).

```
.model small
.stack 100h
.data
string db ' '
```

```
.code
main proc
mov ax,@data
mov ds,ax
mov ah,08h     ;for inputing character without echo
int 21h

   and al,0fh
   mov dh,al
   mov bl,01h
   mov cx,000ah
   again:
   mov al,dh
   mul bl
   aam
   mov bh,al
   cmp ah,00h
   je lable
    or ah,30h
   mov dl,ah
   mov ah,02h
   int 21h
   lable:
   mov al,bh
   or al,30h
   mov dl,al
   mov ah,02h
   int 21h

   mov dl,20h  ;display space after a number
   mov ah,02h
   int 21h
   inc bl
   loop again:
 mov ax,4c00h
 int 21h
main endp
```

14.WAP in 8086 to generate multiplication table of 5 numbers strored in memory as array, store the result and display in following format:(2064 shrawan).

```
5 10 15 20 25 30 35 40 45 50
3 6 9 12 15 18 21 24 27 30
.......................................
```

```
.model small
.stack 64
.data
multiplier db '5','3','4','2','1'   ;5 number stored in memory
.code
main proc
mov ax,@data
mov ds,ax

mov cx,0005h
lea si,multiplier
  nextnum:

   push cx  ;Saving the counter of number
   mov al,[si]
   and al,0fh
   mov dh,al
   mov bl,01h
   mov cx,000ah
   again:
   mov al,dh
   mul bl
   aam
   mov bh,al
   cmp ah,00h
   je lable
   or ah,30h  ;DISPLAYING HIGHER BIT
   mov dl,ah
   mov ah,02h
   int 21h
   lable:
   mov al,bh   ;DISPLAYING LOWER BIT
   or al,30h
   mov dl,al
   mov ah,02h
   int 21h

   mov dl,20h  ;display space after a number
   mov ah,02h
   int 21h
   inc bl
   loop again:

   pop cx       ;Retriving the counnt of the number
   inc si
   dec cl
   cmp cl,00h
   je last

   mov dl,0ah
   mov ah,02h
   int 21h

   mov dl,0dh
   mov ah,02h
   int 21h

   jmp nextnum

 last:
 mov ax,4c00h
 int 21h
main endp
```

15.Write a program to take string input and display the vowel count at the center of the screen.

```
.model small
.stack 100h
.data
kbdat db 21 dup(' '),'$'
count db "Vowel count is:",'$'
```

```
.code
main proc
mov ax,@data
mov ds,ax

mov ah,0ah
lea dx,kbdat
int 21h

lea si,kbdat
mov bl,00h

again:
mov al,[si]
cmp al,'$'     ;Compare for end of string
je a3
 cmp al,0dh  ;Compare for enter
je a3
cmp al,'a'
jb a1
cmp al,'z'
ja a1
sub al,20h    ;Convert to upper case
a1:
cmp al,'A'
jne a4
inc bl
jmp a2
a4:
cmp al,'E'
jnz a5
inc bl
jmp a2
a5:
cmp al,'I'
jnz a6
inc bl
jmp a2
a6:
cmp al,'O'
```

```
jnz a7
inc bl
jmp a2
a7:
cmp al,'U'
jnz a2
inc bl
jmp a2
a2:
inc si
jmp again
a3:



mov ah,02h
mov dh,12
mov dl,40
int 10h

 mov dx,offset count
 mov ah,09h
 int 21h

;converting count to ascii
    mov al,01h
    mul bl
    aam

    mov bh,al
    cmp ah,00h
    je noneed

    add ah,30h
    mov dl,ah
    mov ah,02h
    int 21h

    noneed:
    mov al,bh
```

```
  add al,30h
  mov dl,al
  mov ah,02h
  int 21h


mov ax,4c00h
int 21h
main endp
```