

Assembly Language Program on 8086
Solution for old Questions
2059 to 2066 (Regular and Back ALL)

1. Write an assembly language program to calculate sum of series $1^2 + 2^2 + 3^2 + 4^2 + \dots$ up to ten terms and display the result. [2066 Magh, Regular]

```
.model small
.stack 100
.data
    ten dw 10
    sum dw 00
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov cx,10
    mov dl,01
11:   mov al,dl
    mul dl
    add sum,ax
    inc dl
    loop 11

    mov ax,sum
12:   mov dx,00
    div ten
    inc cx
    add dx,30h
    push dx
    cmp ax,00
    je 13
    jmp 12

13:   pop dx
    mov ah,02h
    int 21h
    loop 13

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

2. Write an assembly language program for 8086 to read a string and find the number of alphabets, numerals and other characters. Display the different counts. [2066 Kartik, Back]
3. Write an assembly language program to get string input; count no. of vowels and display message 'even vowels' on the screen if the count is even otherwise display 'odd vowels'. [2065 Chaitra, Regular]
4. Write down an assembly language program to read a string and display each word on different lines of a clear screen. [2065 kartik, back]

```
.model small
.stack 100h
.data
    string db 50 dup('$')           ;Hold entered string
    sin db "Enter the string:",0ah,0dh,'$'
    sout db "Separate words are:",0ah,0dh,'$'
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h                    ;String length

    mov dx,offset sin            ;message "Enter the string:"
    mov ah,09h
    int 21h
    mov si,offset string          ;Reading the String
next:
    mov ah,01h
    int 21h
    mov [si],al                  ;Character comes in AL
    cmp al,0dh
    je finish
    inc si
    inc bl
    jmp next
```

```
finish:
    mov ax,0600h                ;request scroll
    mov bh,07h                  ;white on black
    mov cx,0000h                ;full screen
    mov dx,184fh
```

```

int 10h

mov dx,offset sout    ;message "Separate words are"
mov ah,09h
int 21h
mov si,offset string
mov ch,00h    ;
mov cl,bl    ;set string length in counter register
up:
mov dl,[si]
cmp dl,' '    ;if equal this is the SPACE
je newline
mov ah,02h
int 21h
jmp below
newline:
mov ah,02h
mov dl,0ah
int 21h
mov dl,0dh
int 21h
below:
inc si
loop up

mov ax,4c00h
int 21h
main endp
end main    ;End program Execution

```

- 5. Write down an assembly language program to read a string and count the no of vowels in the string. Display the no of vowels in the string and the string without the vowels in it in a clear screen with reverse attribute.**

[2064 Poush, regular]

```

.model small
.stack 100h
.data
    abc db 50 dup('$')                ;Hold entered string
    xyz db "Enter the string:","$"
    pqr db "The string is :","$"
    jkl db 0ah,0dh,"The String without vowel is:","$"
    vowel db 0ah,0dh,"No of the Vowel is:","$"
    num db 10 dup('$')
.code

```

```

main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h           ;String length

    mov dx,offset xyz    ;message "Enter the string:"
    mov ah,09h
    int 21h
    mov si,offset abc
;,,,,,,,,; Read string untill press enter key ;,,,,,,,,;

    next:
        mov ah,01h
        int 21h
        mov [si],al      ;Character comes in AL
        cmp al,0dh
        je finish
        inc si
        inc bl
        jmp next
    finish:
        dec si
        mov ch,00h
        mov cl,bl

;,,,,,,,,; To clear the screen ;,,,,,,,,;

        push cx
        mov ax,0600h     ;clear screen and set the attribute
        mov bh,07h       ;white on black
        mov cx,0000h
        mov dx,184fh
        int 10h
        pop cx

;,,,,,,,,; set cursor position in the beginning of the page ;,,,,;

        mov ah,02h
        mov bh,00
        mov dh,00

```

```
mov dl,00
int 10h
```

;;;;;;;;; Display original string ;;;;;;

```
mov dx,offset pqr ;Message for original String
mov ah,09h
int 21h
mov dx,offset abc ;The display of the Original data
mov ah,09h
int 21h
```

;;;;;;;;; To print string without vowel in reverse attribute & count vowels

```
mov dx,offset jkl ; message for the consonant string
mov ah,09h
int 21h
mov dl,00h
count:
mov al,[si]
cmp al,'A'
je increment
cmp al,'E'
je increment
cmp al,'I'
je increment
cmp al,'O'
je increment
cmp al,'U'
je increment
cmp al,'a'
je increment
cmp al,'e'
je increment
cmp al,'i'
je increment
cmp al,'o'
je increment
cmp al,'u'
je increment
push dx
mov dl,al
mov ah,02h
int 21h
pop dx
```

here:

```

        dec si
        loop count
        jmp print
increment:
        inc dl
        mov al,dl
        add al,00h
        daa
        mov dl,al
        jmp here

print:
        mov al,dl
        add al,30h
        mov num,al

;;;;;;;;; Display the no of vowels in the Screen ;;;;;;;;;;

        mov dx,offset vowel
        mov ah,09h
        int 21h
        mov dx,offset num
        int 21h
        mov ax,4c00h      ;Return to the Dos
        int 21h
main endp      ;End procedure
end main      ;End program Execution

```

- 6. Write a program to generate multiplication table of five numbers stored in memory as array, store the result and display in following format.**

5 10 15 20 25 30 35 40 45 50

3 6 9 12 15 18 21 24 27 30

... ..

[2064 Shrawan, back]

```

.model small
.stack 100h
.data
    arr db 5,3,6,4,8
    num dw 00
    ten dw 10
.code
main proc far
    mov ax,@data
    mov ds,ax

```

```

    mov bx,05
    mov si,00
    mov ax,00
11:  mov al, arr[si]
    mov num,ax
    push bx
    mov cx,00
    mov bl,0ah

```

```

12:  mov dx, 0000
    div ten
    inc cx
    add dx, 30h
    push dx
    cmp ax, 00
    je l3
    jmp l2
13:  pop dx
    mov ah, 02h
    int 21h
    loop l3

```

```

    call space
    mov ax,num
    add al,arr[si]
    mov num,ax
    dec bx
    jnz l2
    pop bx
    call newline
    mov ax,00
    mov num,ax
    inc si
    dec bx
    jnz l1

```

```

    mov ax,4c00h
    int 21h
main endp

```

```

space:
    mov dl,' '
    mov ah, 02h
    int 21h
    ret

```

```

newline:
    mov ah, 02
    mov dl, 0ah
    int 21h
    mov ah, 02
    mov dl, 0dh
    int 21h
    ret
end main

```

- 7. Write an assembly language program for 8086 to read string, count the number of vowels in the string and display the string and its vowels count in a clear screen. [2063 Kartik, regular]**

```

.model small
.stack 100h
.data
    abc db 50 dup('$')           ;Hold entered string
    xyz db "Enter the string:",0ah,0dh,'$'
    pqr db "The string is :",'$'
    vowel db "No of the Vowel is:",'$'
    num db 10 dup('$')
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h                 ;String length

    mov dx,offset xyz         ;massage "Enter the string:"
    mov ah,09h
    int 21h

    mov si,offset abc         ;Reading the String
next:
    mov ah,01h
    int 21h
    mov [si],al               ;Character comes in AL
    cmp al,0dh
    je finish
    inc si
    inc bl
    jmp next
finish:

```


;;;;;;;;;; To find the No of the Vowels ;;;;;;;;;;

```
mov si,offset abc
mov ch,00h
mov cl,bl
mov dl,00h
```

count:

```
mov al,[si]
cmp al,'A'
je increment
cmp al,'E'
je increment
cmp al,'I'
je increment
cmp al,'O'
je increment
cmp al,'U'
je increment
cmp al,'a'
je increment
cmp al,'e'
je increment
cmp al,'i'
je increment
cmp al,'o'
je increment
cmp al,'u'
je increment
```

here:

```
inc si
loop count
jmp print
```

increment:

```
inc dl
mov al,dl
add al,00h
daa
mov dl,al
jmp here
```

print: mov num,dl

;;;;;;;;;; Display the no of vowels in the Screen ;;;;;;;;;;

```
mov ax,0600h ;clear screen and set the attribute
```

```

        mov bh,07h           ;white on black
        mov cx,0000h
        mov dx,184fh
        int 10h

        mov dx,offset pqr    ;Message for original String
        mov ah,09h
        int 21h

        call newline
        mov dx,offset abc    ;The display of the Original data
        mov ah,09h
        int 21h

        call newline
        mov dx,offset vowel
        mov ah,09h
        int 21h
        mov al,num
        add al,30h
        mov num,al
        mov dx,offset num
        int 21h

        mov ax,4c00h         ;Return to the Dos
        int 21h
main endp                    ;End procedure

```

```

newline:
        mov ah,02h
        mov dl,0ah
        int 21h
        mov dl,0dh
        int 21h
        ret
end main    ;End program Execution

```

- 8. Write a program 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 background and cyan foreground. [2062 bhadra, regular]**

```

.model small
.stack 100h
.data

```

```

    string db 50 dup('$')           ;Hold entered string
    sin db "Enter the string:",0ah,0dh,'$'
    sout db "Separate words are:",0ah,0dh,'$'
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h                    ;String length

    mov dx,offset sin             ;message "Enter the string:"
    mov ah,09h
    int 21h
    mov si,offset string          ;Reading the String
next:
    mov ah,01h
    int 21h
    mov [si],al                  ;Character comes in AL
    cmp al,0dh
    je finish
    inc si
    inc bl
    jmp next
finish:
    mov ax,0600h                 ;request scroll
    mov bh,1bh                   ;cyan on blue
    mov cx,0000h                 ;full screen
    mov dx,184fh
    int 10h

    call setcursor
    mov dx,offset sout           ;message "Separate words are"
    mov ah,09h
    int 21h
    mov si,offset string
    mov ch,00h                   ;
    mov cl,bl                    ;set string length in counter register
    call setcursor

up:
    mov dl,[si]
    cmp dl,' '                   ;if equal this is the SPACE
    je newline
    mov ah,02h
    int 21h
    jmp below

```

```

newline:
    mov ah,02h
    mov dl,0ah
    int 21h
    mov dl,0dh
    int 21h
    call setcursor

below:
    inc si
    loop up

    mov ax,4c00h
    int 21h
main endp
setcursor:
    mov ah,02h
    mov bh,00h
    mov dh,00    ;row
    mov dl,40    ;center column
    add dh,bl
    int 10h
    inc bl
    ret
end main    ;End program Execution

```

9. Write assembly language program for 8086 to sort five numbers in ascending and descending order. [2062 Baishakh, back]

```

.model small
.stack 100h
.data
    arr db 11,78,52,39,84
    asc db 5 dup('$')
    desc db 5 dup('$')
.code
main proc far
    mov ax,@data
    mov ds,ax

;;;      ascending      ;;;;

    mov dx,4h

```

```

11:  mov cx,dx
    mov si,00h
12:  mov al, arr[si]
    mov bl,arr[si+1]
    mov asc[si],al
    mov asc[si+1],bl
    cmp al,bl
    jc l3
    mov asc[si],bl
    mov asc[si+1],al

13:  inc si
    loop l2
    dec dx
    jnz l1
    ;;;;      descending      ;;;;

    mov dx,4h
14:  mov cx,dx
    mov si,00h
15:  mov al, arr[si]
    mov bl,arr[si+1]
    mov desc[si],al
    mov desc[si+1],bl
    cmp al,bl
    jnc l3
    mov desc[si],bl
    mov desc[si+1],al
16:  inc si
    loop l5
    dec dx
    jnz l4

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

```

10. Write an assembly language program to read a string from the user, convert it to upper case, count the number of words and display each word in each line and number of words. [2061 Ashwin, regular]

```

.model small
.stack 100h
.data
    abc db 50 dup('$')          ;Hold entered string

```

```

xyz db "Enter the string:","$"
pqr db 0ah,0dh,"The uppercase converted string is :","$"
ghi db 0ah,0dh,"The no of words are:","$"
jkl db 0ah,0dh,"The separate words are:",0ah,0dh,'$'
ten dw 10

.code
main proc far
mov ax,@data
mov ds,ax
mov es,ax
mov bl,00h          ;String length

mov dx,offset xyz    ;massage "Enter there"
mov ah,09h
int 21h
mov si,offset abc     ;Reading the String
next1:
mov ah,01h
int 21h
mov [si],al           ;Character comes in AL
cmp al,0dh
je finished
inc si
inc bl
jmp next1
finished:
mov dx,offset pqr     ;massage for converted string
mov ah,09h
int 21h
;;;;;;;;;;;;;To convert in the Upper case;;;;;;;;;;;;;

mov si,offset abc
mov ch,00h
mov cl,bl
next:
mov al,[si]
cmp al,' '
je nochang            ;if equal this is the SPACE
cmp al,60h            ;check for ascii 97 that small 'a'
jb nochang
sub al,20h
nochang:
mov dl,al
mov ah,02h
int 21h
inc si

```

```

        loop next

;,,,,,,,,,,,,, To count the words ;,,,,,,,,,,,,,

        mov dx,offset jkl
        mov ah,09h
        int 21h
        mov si,offset abc
        mov cl,bl
        mov bl,01h
up:
        mov dl,[si]
        cmp dl,' ' ;if equal this is the SPACE
        je newline
        mov ah,02h
        int 21h
        jmp below
newline:
        inc bl
        mov ah,02h
        mov dl,0ah
        int 21h
        mov dl,0dh
        int 21h
below:
        inc si
        loop up

        mov dx,offset ghi
        mov ah,09h
        int 21h
        mov ah,00h
        mov al,bl

11:      mov dx, 0000
        div ten
        inc cx
        add dx, 30h
        push dx
        cmp ax, 00
        je l2
        jmp l1
12:      pop dx
        mov ah, 02h
        int 21h

```

```

        loop l2

        mov ax,4c00h        ;Return to the Dos
        int 21h

main endp        ;End procedure
end main        ;End program Execution

```

11. Write a program to read a string, convert the small case letters to upper case and display the converted string in the next line. [2060 Bhadra, regular]

```

.model small
.stack 100h
.data
    abc db 50 dup('$')        ;Hold entered string
    xyz db "Enter the string:",0ah,0dh,$'
    pqr db "The uppercase converted string is :",0ah,0dh,$'
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h                ;String length

    mov dx,offset xyz        ;message "Enter there"
    mov ah,09h
    int 21h
    mov si,offset abc        ;Reading the String

next1:
    mov ah,01h
    int 21h
    mov [si],al                ;Character comes in AL
    cmp al,0dh
    je finished
    inc si
    inc bl
    jmp next1
finished:
    mov dx,offset pqr        ;message for converted string
    mov ah,09h
    int 21h

```

;;;;;;;;;;;;;To convert in the Upper case;;;;;;;;;;;;;

```

    mov si,offset abc

```



```

        mov ch,00h
        mov cl,bl
next:    mov al,[si]
        cmp al,' '
        je nochang          ;if equal this is the SPACE
        cmp al,60h          ;check for ascii 97 that small 'a'
        jb nochang
        sub al,20h
nochang: mov dl,al
        mov ah,02h
        int 21h
        inc si
        loop next

        mov ax,4c00h        ;Return to the Dos
        int 21h
main endp          ;End procedure
end main           ;End program Execution

```

12. Write an 8086 assembly language program to sort ten 16-bit data stored in a table and display the numbers as decimal numbers in the screen.

[2060 Jestha, back]

```

.model small
.stack 100h
.data
    arr dw 1145h,7898h,5224h,3969h,8422h,4598h,3574h,9526h,5893h,6587h
    ten dw 10
.code
main proc far
    mov ax,@data
    mov ds,ax

;;;          ascending sort          ;;;;

    mov dx,9h
11:  mov cx,dx
    lea bx,arr

```

```

12: mov ax, [bx]
    cmp [bx+2],ax
    jnc 13
    push dx
    mov dx,[bx+2]
    mov [bx+2],ax
    mov [bx],dx
    pop dx
13: add bx,02h
    loop 12
    dec dx
    jnz 11

;;;      display      ;;;;

```

```

    mov cx,0ah
    lea bx,arr
14: mov ax, [bx]
    push cx
    mov cx,00
15: mov dx,0000
    div ten
    inc cx
    add dx,30h
    push dx
    cmp ax,00
    je 16
    jmp 15
16: pop dx
    mov ah,02h
    int 21h
    loop 16
    pop cx
    add bx,02h
    call newline
    loop 14

```

```

    mov ax,4c00h
    int 21h
main endp

```

```

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

```

```

        mov dl, 0dh
        int 21h
        ret
end main

```

13. Write an assembly language program for 8086 to find the largest number among 10 numbers stored as ARR. [2060 Chaitra, back]

```

.model small
.stack 100h
.data
    arr db 11,78,52,39,84,27,71,65,93,57
    result db ?

.code
main proc far
    mov ax,@data
    mov ds,ax
    mov si,offset arr
    mov al,00
    mov cx,0ah
next:
    cmp al,[si]
    jnc below
    mov al,[si]

below:
    inc si
    loop next
    mov result,al
    mov ax,4c00h
    int 21h
main endp
end main

```

14. Write an 8086 assembly language program to separate words from a string. Display each word in separate line. [2059 Shrawan, regular]

```

.model small
.stack 100h
.data
    str db "here is the string."          ;Hold the string
    len dw $-str      ;length of the string
.code

```

```

main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
mov cx,len          ;String length

    mov si,offset str    ;Reading the String
up:
    mov dl,[si]
    cmp dl,' '          ;if equal this is the SPACE
    je newline
    mov ah,02h
    int 21h
    jmp below
newline:
    mov ah,02h
    mov dl,0ah
    int 21h
    mov dl,0dh
    int 21h
below:
    inc si
    loop up

    mov ax,4c00h
    int 21h
main endp
end main    ;End program Execution

```

15. Write an assembly language program in 8086 to read a string, reverse the string and display the original string as well as the reverse string in each line of the cleared screen with red foreground and white background.

[2065 Final Assessment, HCOE]

```

.model small
.stack 100h
.data
    abc db 50 dup('$')          ;Hold entered string
    xyz db "Enter the string:","$"
    pqr db "The string is :","$"
    jkl db 0ah,0dh,"The reverse String:","$"
.code
main proc far
    mov ax,@data
    mov ds,ax
    mov es,ax
    mov bl,00h                ;String length

```

```

    mov dx,offset xyz    ;message "Enter the string:"
    mov ah,09h
    int 21h
    mov si,offset abc

;;;;;;;;;; Read string untill press enter key ;;;;;;;;;;
next:
    mov ah,01h
    int 21h
    mov [si],al          ;Character comes in AL
    cmp al,0dh
    je finish
    inc si
    inc bl
    jmp next
finish:
    dec si                ;To decrease the index of string at the end (lower from newline)

;;;;;;;;;; To clear the screen ;;;;;;;;;;

    mov ax,0600h          ;clear screen and set the attribute
    mov bh,74h            ;red on white
    mov cx,0000h
    mov dx,184fh
    int 10h

;;;;;;;;;; Display original string ;;;;;;;;;;

    mov dx,offset pqr     ;Message for original String
    mov ah,09h
    int 21h
    mov dx,offset abc     ;The display of the Original data
    mov ah,09h
    int 21h

;;;;;;;;;; To print reverse string ;;;;;;;;;;

    mov dx,offset jkl     ; message for the reverse string
    mov ah,09h
    int 21h
    mov ch,00h
    mov cl,bl             ;Total string's length
above:

```

```

    mov al,[si]    ;At first si points to the end
    mov dl,al
    mov ah,02h
    int 21h
    dec si
    loop above

    mov ax,4c00h    ;Return to the Dos
    int 21h
main endp          ;End procedure
end main           ;End program Execution

```