**TASK 1**

.model small

.stack 100h

.data

input db 'Enter your marks : ','$'

Fg db 'GRADE : ','$'

.code

mov ax, @data

mov ds,ax

mov dx, offset input

mov ah, 09h

int 21h

mov dl, 10

mov bl, 0

NUMINPUT:

mov ah, 01h

int 21h

cmp al, 13

je Check

mov ah, 0

sub al, 48

mul dl

mov cl, al

mov al, bl

add al, cl

mov bl, al

jmp NUMINPUT

MOV dl, 10

MOV ah, 02h

INT 21h

MOV dl, 13

MOV ah, 02h

INT 21h

Check:

.if (bl < 50 )

lea dx, Fg

mov ah,09h

int 21h

mov dl,"F"

mov ah,02h

int 21h

.elseif (bl < 59 )

lea dx, Fg

mov ah,09h

int 21h

mov dl,"D"

mov ah,02h

int 21h

.elseif (bl < 75 )

lea dx, Fg

mov ah,09h

int 21h

mov dl,"C"

mov ah,02h

int 21h

.elseif (bl < 90 )

lea dx, Fg

mov ah,09h

int 21h

mov dl,"B"

mov ah,02h

int 21h

.elseif (bl < 100 )

lea dx, Fg

mov ah,09h

int 21h

mov dl,"A"

mov ah,02h

int 21h

.endif

exit:

mov ah, 04ch

int 21h

end

**TASK 2**

.model small

.stack 100h

.data

output db "Enter A String: $"

output1 db "Reversed String: $"

arr1 db 7 dup (?),'$'

temp db 0

.code

MOV AX, @data

MOV DS, AX

jmp main

show proc

mov ah,09h

Int 21h

ret

show endp

takeinput proc

mov si,0

take:

Mov ah, 01h

Int 21h

cmp al,13

JZ done

mov arr1[si],al

inc si

inc temp

jmp take

done:

mov dl,10

mov ah,02h

int 21h

ret

takeinput endp

reverse proc

mov bx,0

mov bl,temp

mov si,offset arr1

inc bl

.while(bl>0)

mov dl,[si+bx]

mov ah,02h

int 21h

dec bl

.endw

mov si,offset arr1

mov dl,[si]

mov ah,02h

int 21h

ret

reverse endp

main :

mov dx, offset output

call show

call takeinput

mov dl,10

mov ah,02h

int 21h

lea dx,offset output1

mov ah,09h

int 21h

call reverse

mov ah,4ch

int 21h

end

**TASK 3**

.model small

.stack 0100h

.data

f1 db "file.txt", 0

buffer db 20 dup('$')

fileinfo dw 0

nam db "Hamza", '$'

.code

mov ax, @data

mov ds, ax

mov ax, 0h

;-------------------------------------------------------------------------

mov ah, 3ch

mov cx, 0

mov dx, offset f1

int 21h

;-------------------------------------------------------------------------

mov ah, 3dh

mov al, 2

mov dx, offset f1

int 21h

;-------------------------------------------------------------------------

mov fileinfo, ax

mov cx, 0

mov dx, 0

mov ah, 42h

mov al, 2

int 21h

;-------------------------------------------------------------------------

mov ah, 40h

mov bx, fileinfo

mov cx, 7

mov dx,offset nam

int 21h

;-------------------------------------------------------------------------

mov ah, 3eh

mov bx, fileinfo

int 21h

;-------------------------------------------------------------------------

mov ah, 3dh

mov al, 0

mov dx, offset f1

int 21h

;-------------------------------------------------------------------------

mov fileinfo, ax

mov ah, 3fh

mov cx, 10

mov dx, offset buffer

mov bx, fileinfo

int 21h

;--------------------------------------------------------------------------

mov dx, offset buffer

mov ah, 09h

int 21h

mov ah, 4ch

int 21h

end