# Muhammad Dawood

21I-1556

# Task 1

dosseg

.model small

.stack 100h

.data

    var1 DB ?

    var2 DB ?

    var3 DW ?

    var4\_a DB ?

    var4\_b DB ?

.code

    main proc

    mov ax,@data

    mov ds,ax

    mov dl,'('

    mov ah,2

    int 21h

    mov ah,1

    int 21h

    sub al,48

    mov var1,al

    mov dl,'\*'

    mov ah,2

    int 21h

    mov ah,1

    int 21h

    sub al,48

    mov var2,al

    mov dl,')'

    mov ah,2

    int 21h

    mov dl,'+'

    mov ah,2

    int 21h

    xor ax,ax

    mov ah,1

    int 21h

    sub al,48

    mov bl, 1

    mul bl

    mov var3,ax

    mov dl,'='

    mov ah,2

    int 21h

    mov al,var1

    mul var2

    mov dx,ax

    add dx,var3

    mov ax,dx

    mov bl,10

    div bl

    add al,48

    add ah,48

    mov var4\_a,al

    mov var4\_b,ah

    mov dl,var4\_a

    mov ah,2

    int 21h

    mov dl,var4\_b

    mov ah,2

    int 21h

    mov ah,4ch

    int 21h

main endp

end main

# Task 2

;two numbers are equal or not

dosseg

.model small

.stack 100h

.data

var1 BYTE ?

var2 BYTE ?

Msg1 db 'numbers are equal$'

Msg2 db 'numbers are not equal$'

.code

main proc

Mov ax,@data

Mov ds,ax

mov ah, 1

int 21h

mov var1, al

Mov dl,al

Mov ah,1

Int 21h

mov var2, al

Cmp dl,al

Je l1

    Mov dx, offset Msg2

    Mov ah,9

    Int 21h

Mov ah,4ch

Int 21h

l1:

mov dx, offset Msg1

mov ah,9

int 21h

mov ah,4ch

int 21h

main endp

end main

# Task 3

; Print characters from A to Z

dosseg

.model small

.stack 100h

.data

.code

main proc

mov cl,26

mov dl,65

l1:

mov ah,2

int 21h

inc dl

loop l1

mov dl, 13

mov ah, 2

int 21h

mov dl, 10

mov ah, 2

int 21h

xor cx, cx

xor dx, dx

mov cl, 26

mov dl, 91

l2:

sub dl, 1

mov ah, 2

int 21h

loop l2

mov ah,4ch

int 21h

main endp

end main

# Task 4

;determine even or odd

dosseg

.model small

.stack 100h

.data

Msg1 db 'number is even$'

Msg2 db 'number is odd$'

.code

main proc

Mov ax,@data

Mov ds,ax

xor ax, ax

mov ah, 1

int 21h

mov bl, 2

div bl

cmp ah, 0

je l1

Mov dx, offset Msg2

    Mov ah,9

    Int 21h

mov ah, 4ch

int 21h

l1:

mov dx, offset Msg1

mov ah,9

int 21h

mov ah, 4ch

int 21h

main endp

end main

# Task 5

;table of 3

dosseg

.model small

.stack 100h

.data

var1 db 3

var2 db 1

msg db '3 \* $'

msg2 db ': $'

var3 db 0

var4 db 0

.code

main proc

Mov ax,@data

Mov ds,ax

mov cl,10 ; inc cl for each iteration

l1:

mov al, 3 ; mov 3 to al each time because multiplication result is stored in ax

mov bl,var2 ; this contains the 1 to 10 iteration

mul bl ; multiply al to bl and store result in ax

mov dx,ax ; moving ax into dx because dx is divided by 10 to get a remainder and quotient for printing

mov bl,10 ; divide dx by bl

div bl ; divide dx by bl

mov var3,ah

mov var4,al

add var3,48

add var4,48

mov dl,10

mov ah,2

int 21h

Mov dx, offset msg

Mov ah,9

Int 21h

mov dl,var2

add dl,48

mov ah,2

int 21h

Mov dx, offset msg2

Mov ah,9

Int 21h

mov dl,var4

mov ah,2

int 21h

mov dl,var3

mov ah,2

int 21h

add var2, 1

loop l1

mov ah,4ch

int 21h

main endp

end main