**TASK 1**

.model small

.stack 010h

.data

var1 db 0

var2 db 0

var3 db 0

a1 db "Triangle Is Equilateral"

a2 db "Triangle Is Isosceles"

a3 db "Triangle Is Scalene"

.code

mov ax,@data

mov ds,ax

mov ax,0

mov ah,01h

int 21h

sub al,48

mov var1,al

mov ah,01h

int 21h

sub al,48

mov var2,al

mov ah,01h

int 21h

sub al,48

mov var3,al

;========================================

mov cl,var2

cmp var1,cl

je check1

mov cl,var3

cmp var2,cl

je check2

cmp cl,var1

je check2

mov cx,lengthof a3

mov si,offset a3

l3:

mov dl,[si]

mov ah,02h

int 21h

inc si

loop l3

jmp exit

;========================================

check1:

mov cl,var3

cmp var2,cl

jne check2

mov cx,lengthof a1

mov si,offset a1

l1:

mov dl,[si]

mov ah,02h

int 21h

inc si

loop l1

jmp exit

check2:

mov cx,lengthof a2

mov si,offset a2

l2:

mov dl,[si]

mov ah,02h

int 21h

inc si

loop l2

jmp exit

exit:

mov ah,4ch

int 21h

end

**TASK 05**

.MODEL SMALL

.STACK 100H

.DATA

STAR DB ?

BLANK DB ?

.CODE

MAIN PROC

mov ah,01h

int 21h

mov ah,0

mov cx,ax

MOV BX,1

L1:

PUSH CX

L2:

MOV AH,2

MOV DL,32

INT 21H

LOOP L2

MOV CX,BX

L3:

MOV AH,2

MOV DL,'\*'

INT 21H

LOOP L3

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

INC BX

INC BX

POP CX

LOOP L1

DEC al

MOV CX,ax

MOV BH,7

MOV BL,2

MOV STAR,BH

MOV BLANK,BL

L4:

CMP BLANK,0

JE L5

MOV AH,2

MOV DL,32

INT 21H

DEC BLANK

;CMP BLANK,0

JMP L4

L5:

MOV AH,2

MOV DL,'\*'

INT 21H

DEC STAR

CMP STAR,0

JNE L5

L6:

MOV AH,2

MOV DL,10

INT 21H

MOV DL,13

INT 21H

DEC BH

DEC BH

MOV STAR,BH

INC BL

MOV BLANK,BL

LOOP L4

EXIT:

MOV AH,4CH

INT 21H

MAIN ENDP

END MAIN

**TASK 03**

.model small

.stack 100h

.data

val1 db 0

val2 db 0

val3 db 0

result db 0

.code

main proc

mov ax, @data

mov ds, ax

mov ax, 0

mov ah,01h

int 21h

mov ah,0

sub ax,48

mov val1,al

mov ah,01h

int 21h

mov ah,0

sub ax,48

mov val2,al

mov ah,01h

int 21h

mov ah,0

;sub ax,48

mov val3,al

cmp val3,2bh

je addd

cmp val3,2ah

je mulll

cmp val3,2dh

je subbb

cmp val3,2fh

je divvv

addd:

call addition

jmp exit

subbb:

call subtraction

jmp exit

mulll:

call multiplication

jmp exit

divvv:

call division

jmp exit

main endp

addition proc

mov bl,val1

mov cl,val2

add bl,cl

mov result,bl

mov dl,result

add dl,48

mov ah,02h

int 21h

ret

jmp exit

addition endp

subtraction proc

mov bl,val1

mov cl,val2

sub bl,cl

mov result,bl

mov dl,result

add dl,48

mov ah,02h

int 21h

ret

jmp exit

subtraction endp

multiplication proc

mov ah,0

mov al,val1

mov bl,val2

mul bl

mov result,al

mov dl,result

add dl,48

mov ah,02h

int 21h

ret

jmp exit

multiplication endp

division proc

mov al,val1

mov bl,val2

div bl

mov result,bl

mov dl,result

add dl,48

mov ah,02h

int 21h

ret

jmp exit

division endp

exit:

mov ah,4ch

int 21h

end

**TASK 02**

.model small

.stack 100h

.data

.code

jmp main

sum proc

mov cl, bl

mov bx, 0

mov bl, 2

div bl

cmp ah, 0

je J

dec cl

J:

mov ax, 0

mov bx, 0

L:

add bl, cl

dec cx

loop L

ret

sum endp

main proc

mov ax, @data

mov ds, ax

mov ax, 0

mov ah, 01h

int 21h

mov ah, 0

sub al, 48

mov bl, 10

mul bl

mov bl, al

mov ax, 0

mov ah, 01h

int 21h

sub al, 48

add bl, al

mov ax, 0

mov al, bl

call sum

mov ax, 0

mov al, bl

mov bx, 0

mov bl, 100

div bl

mov cl, ah

mov dl, al

add dl, 48

mov ah, 02h

int 21h

mov bx, 0

mov al, cl

mov ah, 0

mov bx, 0

mov bl, 10

div bl

mov cl, ah

mov dl, al

add dl, 48

mov ah, 02h

int 21h

mov dl, cl

add dl, 48

mov ah, 02h

int 21h

mov ah, 4ch

int 21h

main endp

end

**TASK 06**

.model small

.stack 100h

.data

arr db 10h,9,5,20h,4

.code

mov ax, @data

mov ds, ax

mov si, offset arr

mov cx, 5

mov bl, [si]

L1:

cmp [si], bl

jge large

comp:

inc si

loop L1

add bl, 48

mov dl, bl

mov ah, 2

int 21h

large:

mov bl, [si]

jmp comp