**TASK 1**

.model small

.stack 100h

.data

arr db ":--> key Is Not Pressed",'$'

arr1 db ":-->key Is Pressed",'$'

.code

mov ax,@data

mov ds,ax

mov ax,0

L1:

mov ah,01

int 16h

jnz exit

lea dx,arr

mov ah,09h

int 21h

mov dl,10

mov ah,02h

int 21h

jmp L1

exit:

mov dl,10

mov ah,02h

int 21h

lea dx,arr1

mov ah,09h

int 21h

mov ah,4ch

int 21h

end

**TASK 2**

.model small

.stack 100h

.data

arr db ":--> key Is Not Pressed",'$'

arr1 db ":-->key Pressed Is ",'$'

var db (0)

.code

mov ax,@data

mov ds,ax

mov ax,0

L1:

mov ah,01

int 16h

mov var,al

jnz exit

lea dx,arr

mov ah,09h

int 21h

mov dl,10

mov ah,02h

int 21h

jmp L1

exit:

mov dl,10

mov ah,02h

int 21h

lea dx,arr1

mov ah,09h

int 21h

mov dl,var

mov ah,02h

int 21h

mov ah,4ch

int 21h

end

**TASK 3**

.model small

.stack 100H

.data

var db (0)

.code

mov ax,@data

mov ds,ax

mov ax,0

mov cx,10

mov dx,10

mov ah,00h

mov al,13h

int 10h

mov ah,0ch

mov al,14

int 10h

l1:

mov ah,00

int 16h

cmp ah,48h

je up

cmp ah,50h

je down

cmp ah,4dh

je right

cmp ah,4bh

je left

cmp al,27

je exit

up:

sub dx,1

mov ah,0ch

mov al,14

int 10h

jmp l1

down:

add dx,1

mov ah,0ch

mov al,14

int 10h

jmp l1

right:

add cx,1

mov ah,0ch

mov al,14

int 10h

jmp l1

left:

sub cx,1

mov ah,0ch

mov al,14

int 10h

jmp l1

exit:

mov ah, 4ch

int 21H

end

TASK 4

.model small

.stack 100h

.data

string db "H","$"

.code

mov ax,@data

mov ds,ax

mov ax,0

mov ah, 0

mov al, 13h

int 10h

MOV DX, 20

L1:

MOV CX, 20

L2:

MOV AL, 14

MOV AH, 0CH

INT 10H

INC CX

CMP CX,60

JNE L2

INC DX

CMP DX,60

JNE L1

MOV DX, 20

L3:

MOV CX, 70

L4:

MOV AL, 14

MOV AH, 0CH

INT 10H

INC CX

CMP CX,110

JNE L4

INC DX

CMP DX,60

JNE L3

MOV DX, 20

L5:

MOV CX, 120

L6:

MOV AL, 14

MOV AH, 0CH

INT 10H

INC CX

CMP CX,160

JNE L6

INC DX

CMP DX,60

JNE L5

mov dx,20

L7:

MOV CX, 170

L8:

MOV AL, 14

MOV AH, 0CH

INT 10H

INC CX

CMP CX,210

JNE L8

INC DX

CMP DX,60

JNE L7

mov cx,4

ll1:

mov ah,02h

mov dl,10

int 21h

loop ll1

mov dl,"A"

int 21h

int 10h

mov ah,4ch

int 21h

end