.MODEL SMALL

.STACK 100H

.DATA

a db "invalid input! Try again.$"

b db "Enter a number between 5 and 9: $"

c db "Please give your inputs: $"

d db "Extracted digits are: $"

e db "The squares are: $"

f db ",$"

arr1 db 4 dup(?)

.CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

; YOUR CODE STARTS HERE

;input

lea dx,b

mov ah,09h

int 21h

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

mov ah,1

int 21h

mov bl, al

sub bl, 30h

cmp bl, 5 ;compair with 5

jg j1

jmp e1

e1: ;invalid number

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

lea dx,a

mov ah,09h

int 21h

jmp exit

j1:

cmp bl, 9 ;compair with 9

jl j2

jmp e1

j2: ;valid number

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

mov cl, bl

jmp exit

exit: ;end of task (a)

lea dx,c

mov ah,09h

int 21h

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

mov bh,cl ;coping the input for farther tasks

add bh,30h

mov bl,30h

mov cx,4 ;task 2 & 3 starts[set cx to 4 and si to 0 for arr\_of\_num]

mov si,0

loop1: ;loop for taking inputs

cmp bl, bh

jge arrays ;after taking all the inputs goes to output the array

mov ah,01h

int 21h

add bl,1

cmp al, 30h ;taking all the numbers and avoiding comma

jge arr\_of\_num

jmp comma

comma: ;an empty condition for comma

jmp loop1

arr\_of\_num: ;array for numbers are generated here

cmp al, 39h

jg comma

mov arr1[si], al

inc si

loop loop1

arrays: ;printing the new found array only using a condition and a loop

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

mov si, 0

mov cx,4

lea dx,d

mov ah,09h

int 21h

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

loop2: ;loop starts from here

mov dl,arr1[si]

mov ah,02h

int 21h

MOV AH ,2

MOV DL,0DH

INT 21h

MOV DL,0AH

INT 21h

inc si

loop loop2

call sq\_gen

; YOUR CODE ENDS HERE

MOV AX, 4C00H

INT 21H

MAIN ENDP

sq\_gen proc

mov ah,09h

lea dx,e

int 21h

mov si,0

mov cx,4

mov ax,0

mov bx,0

loop3:

mov al,arr1[si]

sub al,30h

mov bl,arr1[si]

sub bl,30h

mul bl

mov ah,02h

mov dl, al

add dl, 30h

int 21h

inc si

mov ah,09h

lea dx,f

int 21h

loop loop3

ret

sq\_gen endp

END MAIN