**OSAMA SAGHEER**

**BCS211074**

**SECTION 1**

**TASK 6**

**TASK 1**

**CODE:**

.model small

.data

msg db "A$"

.code

main proc

mov ah, 02h ; set function to print character

mov cx, 26 ; set loop counter to 26 (for A to Z)

mov dl, 'A' ; set initial character to 'A'

mov ah, 02h ; set function to print character

l1:

mov msg, dl ; move character to message buffer

int 21h ; print character

inc dl ; increment character

loop l1 ; jump back to l1 until cx=0

mov ah, 4ch ; set function to exit program

int 21h ; call DOS to exit program

main endp

end main

**OUTPUT:**

****

**TASK 2:**

**CODE:**

.model small

.data

msg db "a$"

.code

main proc

mov ah, 02h ; set function to print character

mov cx, 26 ; set loop counter to 26 (for a to z)

mov dl, 'a' ; set initial character to 'a'

mov ah, 02h ; set function to print character

l1:

mov msg, dl ; move character to message buffer

int 21h ; print character

inc dl ; increment character

loop l1 ; jump back to l1 until cx=0

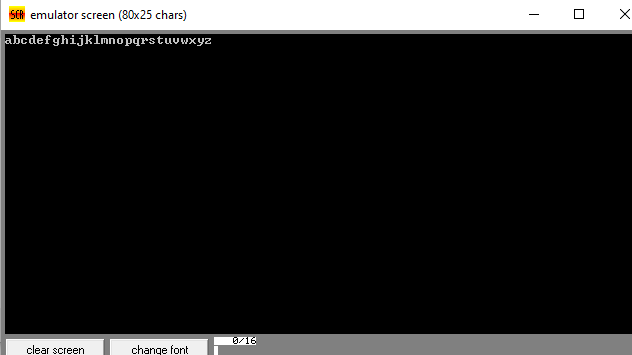
mov ah, 4ch ; set function to exit program

int 21h ; call DOS to exit program

main endp

end main

**OUTPUT:**



**TASK 3:**

**CODE:**

.model small

.data

msg db 10, 13, 'Enter a number: $'

max db 10, 13, 'The highest number is: $'

num db ?

highest db ?

count db 10

.code

main proc

mov ax, @data

mov ds, ax

mov bl, 0 ; initialize highest value to 0

loop\_start:

mov ah, 9 ; print message to enter a number

lea dx, msg

int 21h

; read the number

mov ah, 1 ; input function

int 21h

; store the number

mov num, al

; compare the number with the current highest value

cmp num, bl

jle skip

; if the number is greater than the current highest value, update highest value

mov bl, num

mov highest, bl

skip:

; decrement the count variable

dec count

; check if we have read 10 numbers

cmp count, 0

jne loop\_start

; print the highest value

mov ah, 9 ; print message for highest value

lea dx, max

int 21h

; print the highest value

mov dl, highest

add dl, 48 ; convert the number to its ASCII equivalent

mov ah, 2 ; output function

int 21h

; exit

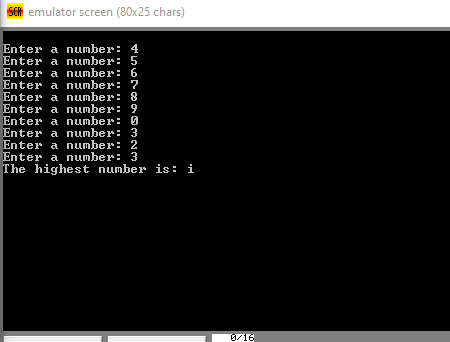
mov ah, 4ch

int 21h

main endp

end main

**OUTPUT:**

****

**TASK 4:**

**CODE:**

.model small

.data

num db 9

.code

main proc

mov ax, @data ; initialize data segment

mov ds, ax

l1:

mov dl, num ; move number to dl register

add dl, '0' ; convert number to character

mov ah, 02h ; set function to print character

int 21h ; print character

dec num ; decrement number

cmp num, 0 ; compare number to 0

jge l1 ; jump back to l1 until num < 0

mov ah, 4ch ; set function to exit program

int 21h ; call DOS to exit program

main endp

end main

**OUTPUT:**

****

**TASK 5:**

**CODE:**

.model small

.data

num db 9

.code

main proc

mov ax, @data ; initialize data segment

mov ds, ax

l1:

mov dl, num ; move number to dl register

add dl, '0' ; convert number to character

test dl, 1 ; check if number is odd

jz l2 ; if even, jump to l2

mov ah, 02h ; set function to print character

int 21h ; print character

dec num ; decrement number

cmp num, 0 ; compare number to 0

jge l1 ; jump back to l1 until num < 0

l2:

dec num ; decrement number

cmp num, 0 ; compare number to 0

jge l1 ; jump back to l1 until num < 0

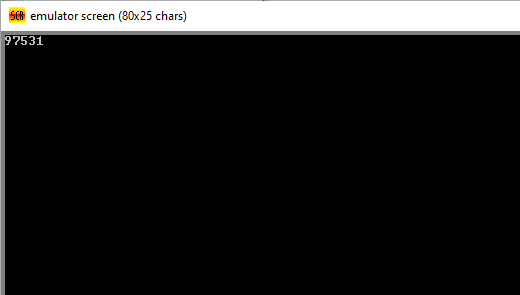
mov ah, 4ch ; set function to exit program

int 21h ; call DOS to exit program

main endp

end main

**OUTPUT:**

****

**TASK 6:**

**CODE:**

.model small

.data

num db 8

.code

main proc

mov ax, @data ; initialize data segment

mov ds, ax

l1:

mov dl, num ; move number to dl register

add dl, '0' ; convert number to character

test dl, 1 ; check if number is odd

jnz l2 ; if odd, jump to l2

mov ah, 02h ; set function to print character

int 21h ; print character

dec num ; decrement number

cmp num, 0 ; compare number to 0

jge l1 ; jump back to l1 until num < 0

l2:

dec num ; decrement number

cmp num, 0 ; compare number to 0

jge l1 ; jump back to l1 until num < 0

mov ah, 4ch ; set function to exit program

int 21h ; call DOS to exit program

main endp

end main

**OUTPUT:**

