**BSAI - LAB TASK**

1. Write a program to calculate the area of various geometric shapes (circle, triangle, and rectangle) based on user input.

**Code:**

.model small

.stack 100h

.data

choice\_msg db 'Choose a shape:',13,10

db '1. Circle',13,10

db '2. Triangle',13,10

db '3. Rectangle',13,10

db 'Enter your choice (1/2/3): $'

radius\_msg db 'Enter radius of circle (in cm): $'

base\_msg db 'Enter base of triangle (in cm): $'

height\_msg db 'Enter height of triangle (in cm): $'

length\_msg db 'Enter length of rectangle (in cm): $'

width\_msg db 'Enter width of rectangle (in cm): $'

area\_msg db 'Area = $'

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 09h

lea dx, choice\_msg

int 21h

input\_choice:

mov ah, 01h

int 21h

sub al, 30h ; Convert ASCII to integer

cmp al, '1'

je calculate\_circle

cmp al, '2'

je calculate\_triangle

cmp al, '3'

je calculate\_rectangle

jmp input\_choice

calculate\_circle:

mov ah, 09h

lea dx, radius\_msg

int 21h

call get\_input

mov bx, ax

mov ax, bx

mul bx

mov cx, 314 ; Value of pi (3.14) multiplied by 100 for precision

mul cx

mov bx, 100

div bx ; Divide by 100 to get the final result

call display\_area

jmp exit\_program

calculate\_triangle:

mov ah, 09h

lea dx, base\_msg

int 21h

call get\_input

mov bx, ax

mov ah, 09h

lea dx, height\_msg

int 21h

call get\_input

add ax, bx

mul bx

mov ax, ax

mov bx, 2

div bx

call display\_area

jmp exit\_program

calculate\_rectangle:

mov ah, 09h

lea dx, length\_msg

int 21h

call get\_input

mov bx, ax

mov ah, 09h

lea dx, width\_msg

int 21h

call get\_input

mul bx

call display\_area

jmp exit\_program

get\_input:

mov ah, 01h

int 21h

sub al, 30h ; Convert ASCII to integer

mov bl, al

mov ax, 0

input\_loop:

mov ah, 01h

int 21h

cmp al, 13 ; Check for carriage return

je input\_done

sub al, 30h ; Convert ASCII to integer

mov cl, al

mov al, bl

mul bx

mov bl, 10

add ax, cx

jmp input\_loop

input\_done:

ret

display\_area:

mov ah, 09h

lea dx, area\_msg

int 21h

mov si, 10

convert\_and\_display:

xor dx, dx

div si

add dl, 30h

push dx

cmp ax, 0

jz display\_loop

jmp convert\_and\_display

display\_loop:

pop dx

mov ah, 02h

int 21h

loop display\_loop

jmp exit\_program

exit\_program:

mov ax, 4C00h

int 21h

main endp

end main

1. Design an assembly language program to convert temperature from Celsius to Fahrenheit or vice versa.

**Code:**

.model small

.stack 100h

.data

choice\_msg db 'Choose conversion:',13,10

db '1. Celsius to Fahrenheit',13,10

db '2. Fahrenheit to Celsius',13,10

db 'Enter your choice (1/2): $'

temp\_msg db 'Enter temperature: $'

result\_msg db 'Result = $'

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 09h

lea dx, choice\_msg

int 21h

input\_choice:

mov ah, 01h

int 21h

sub al, 30h ; Convert ASCII to integer

cmp al, '1'

je celsius\_to\_fahrenheit

cmp al, '2'

je fahrenheit\_to\_celsius

jmp input\_choice

celsius\_to\_fahrenheit:

mov ah, 09h

lea dx, temp\_msg

int 21h

call get\_input

mov bx, ax

mov ax, bx

imul bx, 9

mov cx, 5

idiv cx

add ax, 32

call display\_result

jmp exit\_program

fahrenheit\_to\_celsius:

mov ah, 09h

lea dx, temp\_msg

int 21h

call get\_input

mov bx, ax

sub bx, 32

mov ax, bx

imul bx, 5

mov cx, 9

idiv cx

call display\_result

jmp exit\_program

get\_input:

mov ah, 01h

int 21h

sub al, 30h ; Convert ASCII to integer

mov bl, al

mov ax, 0

input\_loop:

mov ah, 01h

int 21h

cmp al, 13 ; Check for carriage return

je input\_done

sub al, 30h ; Convert ASCII to integer

mov cl, al

mov al, bl

mul bx

mov bl, 10

add ax, cx

jmp input\_loop

input\_done:

ret

display\_result:

mov ah, 09h

lea dx, result\_msg

int 21h

mov si, 10

convert\_and\_display:

xor dx, dx

div si

add dl, 30h

push dx

cmp ax, 0

jz display\_loop

jmp convert\_and\_display

display\_loop:

pop dx

mov ah, 02h

int 21h

loop display\_loop

jmp exit\_program

exit\_program:

mov ax, 4C00h

int 21h

main endp

end main

1. Implement an assembly language program to check if a given number is prim or not.

**Code:**

.model small

.stack 100h

.data

num\_msg db 'Enter a number: $'

prime\_msg db 'The number is PRIME.',13,10,'$'

db 'The number is NOT PRIME.',13,10,'$'

.code

main proc

mov ax, @data

mov ds, ax

mov ah, 09h

lea dx, num\_msg

int 21h

call get\_input

mov bx, ax

call is\_prime

cmp bx, 0

je not\_prime

mov ah, 09h

lea dx, prime\_msg

int 21h

jmp exit\_program

not\_prime:

mov ah, 09h

lea dx, prime\_msg

int 21h

exit\_program:

mov ax, 4C00h

int 21h

main endp

get\_input:

mov ah, 01h

int 21h

sub al, 30h ; Convert ASCII to integer

mov bl, al

mov ax, 0

input\_loop:

mov ah, 01h

int 21h

cmp al, 13 ; Check for carriage return

je input\_done

sub al, 30h ; Convert ASCII to integer

mov cl, al

mov al, bl

mul bx

mov bl, 10

add ax, cx

jmp input\_loop

input\_done:

ret

is\_prime:

mov cx, 2

check\_prime\_loop:

mov dx, 0

div cx

cmp dx, 0

je not\_prime\_exit

inc cx

cmp cx, ax

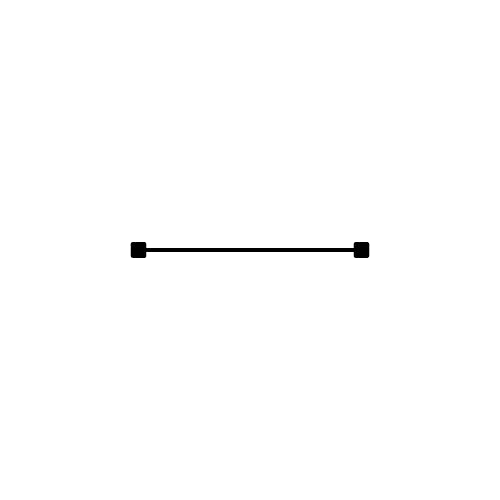
jb check\_prime\_loop

mov bx, 1

ret

not\_prime\_exit:

xor bx, bx

**** ret

end main