**COAL LAB 08 (LAB TASKS)**

**TASK # 01:**

**CODE:**

TITLE TASK1 (test.asm)

INCLUDE IRVINE32.INC

.data

value dword ?

message byte "Genrating 20 random integers between 0 and 998:",0

.code

main proc

mov eax, white + (green \* 16)

call SetTextColor

mov edx, offset message ; to print the prompt message

call WriteString

call crlf

mov dl,0 ; to access column to gotoxy

mov dh,1 ; to access row to gotoxy

mov ecx, 20

call randomize

l1:

mov eax, 990 ; setting the range

call randomrange

mov value, eax

mov eax, 5

call delay

mov eax, value

call gotoxy

call WriteDec

inc dh ; moving to next row

inc dl ; moving to next column

inc dl

loop l1

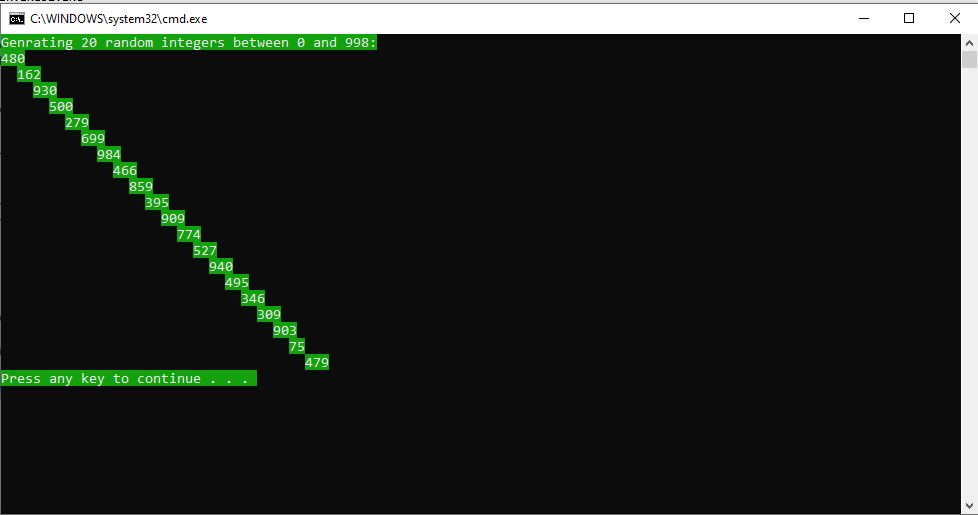
call crlf

exit

main endp

end main

**OUTPUT:**

****

**TASK # 02:**

**CODE:**

TITLE TASK2 (test.asm)

INCLUDE IRVINE32.INC

.data

message byte "Enter any integer : ",0

text1 byte "The number you entered : ",0

text2 byte "The number in binary form : ",0

text3 byte "The number in hexa form : ",0

var1 byte ?

.code

main proc

mov edx, offset message

call WriteString

call readInt

mov var1, al

mov edx, offset text1

call WriteString

call writeDec

call crlf

mov edx, offset text2

call WriteString

call writeBin

call crlf

mov edx, offset text3

call WriteString

call writeHex

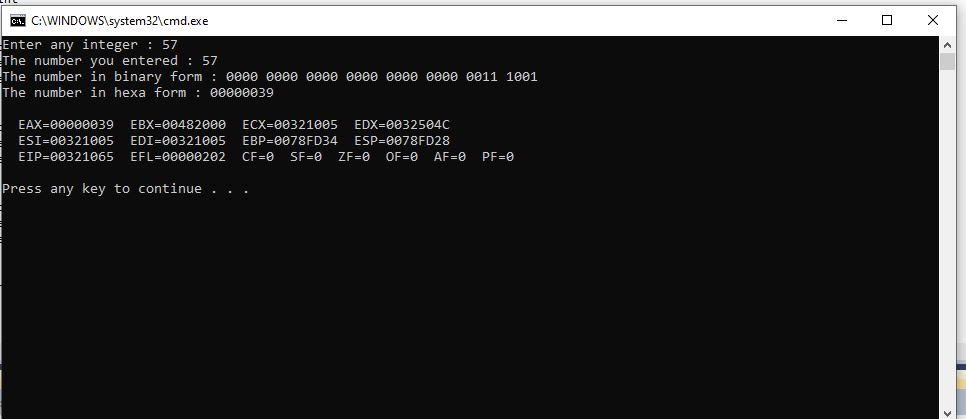
call crlf

call dumpregs

exit

main endp

end main

**OUTPUT:**

**TASK # 03:**

**CODE:**

TITLE TASK3 (test.asm)

INCLUDE IRVINE32.INC

.data

message1 byte "Enter your Employee ID : ",0

message2 byte "Enter your Name : ",0

message3 byte "Enter your Year of Birth : ", 0

message4 byte "Enter your Annual Salary : ",0

TaxMessage1 byte "KINDLY PAY YOUR TAXES! : ",0

TaxMessage2 byte "WHOO, NO TAXES DUE!",0

TaxMessage byte "Your Total Tax : ",0

ID word ?

N byte 30 Dup(?)

Year word ?

Salary dword ?

TAX dword ?

.code

main proc

mov edx, offset message1

call writeString

call readint

mov ID, ax

mov edx, offset message2

call writeString

mov ecx, sizeof N

mov edx, offset N

call ReadString

;mov N, eax

mov edx, offset message3

call writeString

call readint

mov Year, ax

mov edx, offset message4

call writeString

call readint

mov Salary, eax

mov ebx, 50000d

cmp eax, ebx

jc l2

jz l2

mov edx, 0

mov ecx, 0

cmp [Salary+2],0

jnz l1

jmp ln

l1:

xchg dx, word ptr [Salary+2]

mov [Salary+2], 0

mov eax, Salary

ln:

mov cx, 12

div cx

mov edx, 0

mov cx, 2

div cx

mov TAX, eax

jmp l3

l2: mov TAX, 0

mov ebx, 0

mov edx, offset TaxMessage2

call MsgBox

jmp l4

l3:

mov eax, TAX

mov ebx, 0

mov edx, offset TaxMessage1

call MsgBox

l4:

call crlf

mov edx, offset TaxMessage

call writeString

mov eax, TAX

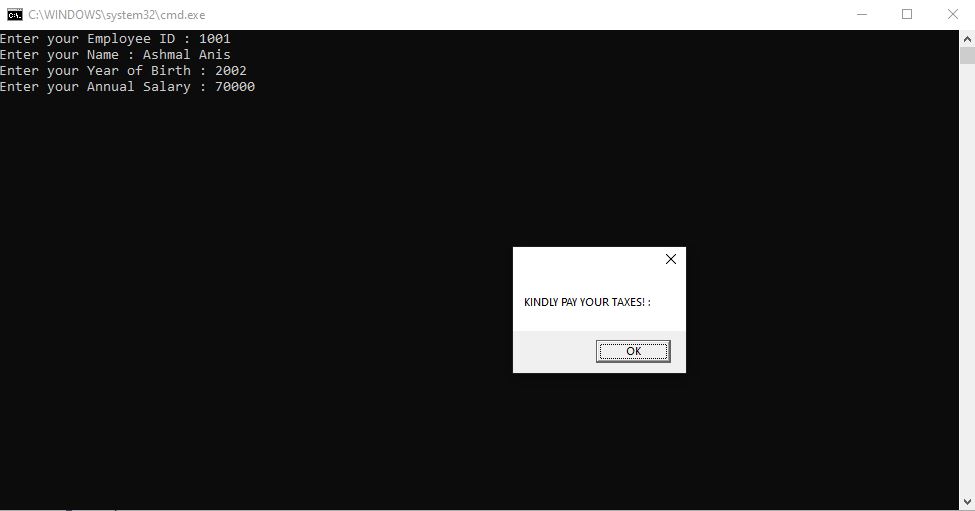
call WriteDec

call crlf

exit

main endp

end main

**OUTPUT:**

**TASK # 04:**

**CODE:**

TITLE TASK4 (test.asm)

INCLUDE IRVINE32.INC

.data

T1 word 0

T2 word 1

nextTerm word ?

fabb word 8 dup(?)

fabbAns byte "0 1 1 2 3 5 8 13",0

filehandle DWORD ?

filename BYTE "Fabonacci.txt", 0

message1 byte "OUTPUT ON CONSOLE",0

message2 byte "OUTPUT STORED IN FILE TOO, PLEASE CHECK Fabonacci.txt",0

.code

main proc

mov esi, 0

mov ax, T1

mov fabb[esi], ax

add esi, 2

mov ax, T2

mov fabb[esi], ax

add esi, 2

mov ecx, 6

L1: mov bx, T1

add bx, T2

mov nextTerm, bx

mov dx, T2

mov T1, dx

mov dx, nextTerm

mov T2, dx

mov eax, 0

mov ax, nextTerm

mov fabb[esi], ax

add esi, 2

loop L1

mov esi, 0

mov ecx, lengthof fabb

mov edx, offset message1

call writeString

call crlf

l2:

mov ax, fabb[esi]

add esi, 2

call writeDec

call crlf

loop l2

mov edx, offset filename

call CreateOutputFile

mov filehandle, eax

mov eax, filehandle

mov edx, OFFSET fabbAns

mov ecx, lengthof fabbAns

call WriteToFile

mov edx, offset message2

call writeString

call crlf

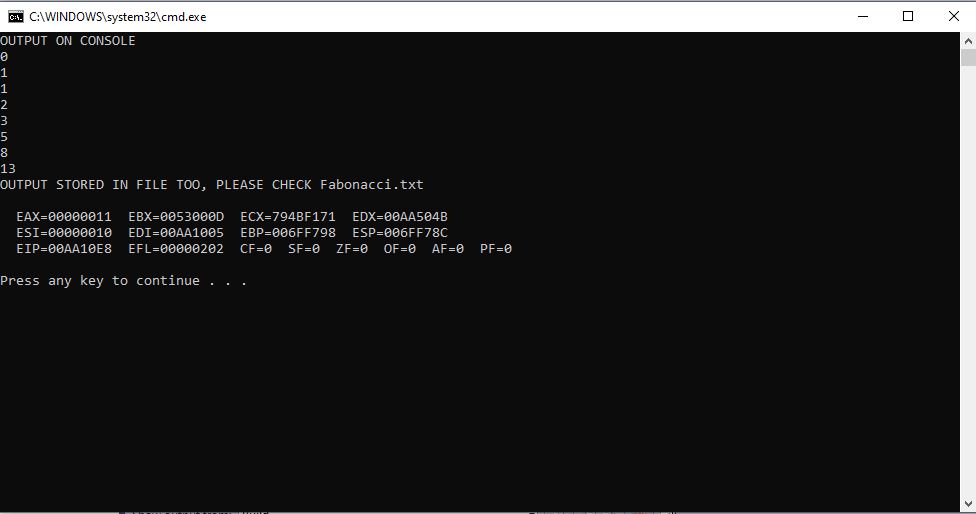
call dumpregs

exit

main endp

end main

**OUTPUT:**

****

**TASK # 05:**

**CODE:**

TITLE TASK5 (test.asm)

INCLUDE IRVINE32.INC

.data

star byte "\*",0

count dword ?

count2 dword 1

x byte 0

y byte 4

.code

main proc

mov ecx, 5

l1:

mov count, ecx

mov ecx, count2

mov dh, x

mov dl, y

call gotoxy

l2:

mov edx, offset star

call WriteString

loop l2

inc x

dec y

mov ecx, count

inc count2

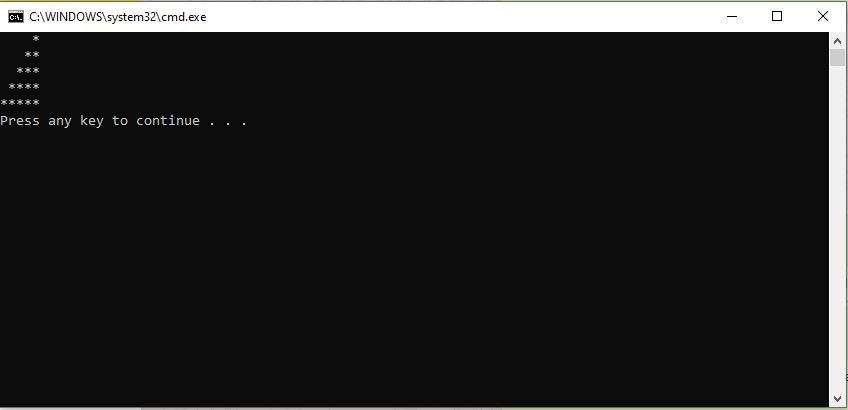
loop l1

call crlf

exit

main endp

end main

**OUTPUT:**