A picture containing text

Description automatically generated

**Name: Saad Rehman  
Student ID: 21F-9640  
Section: 3A**  
**Degree: BS-CS  
Teacher Name: Sir Abdul Qadeer Bilal  
Course: EL-2003 (COAL)  
Lab Number: 09  
……………………………………………………………………………**

**Note: Perform each Question on AFD Debugger.**

**Qno.1**

Write a program which will ask the user to enter his/her marks (out of 100). Define a subroutine that

will display grades according to the marks entered as below:

Marks        Grade

91-100         AA

81-90          AB

71-80          BB

61-70          BC

51-60          CD

41-50          DD

&lt;=40          Fail

**Answer**

[org 0x0100]

jmp start

grade: dw 0

displayGrade: ;displayGrade as a label

cmp ax,90 ;comparing if number is 90 then it would be AA grade

jA AA

cmp ax,80 ;comparing if number is 80 then it would be AB grade

jA AB

cmp ax,70 ;comparing if number is 70 then it would be BB grade

jA BB

cmp ax,60 ;comparing if number is 60 then it would be BC grade

jA BC

cmp ax,50 ;comparing if number is 50 then it would be CD grade

jA CD

cmp ax,40 ;comparing if number is 40 then it would be D grade

jA D

cmp ax,40

mov word[grade],29

ret

AA:

mov word [grade],91

ret

AB:

mov word [grade],81

ret

BB:

mov word [grade],71

ret

BC:

mov word [grade],61

ret

CD:

mov word [grade],51

ret

D:

mov word [grade],41

ret

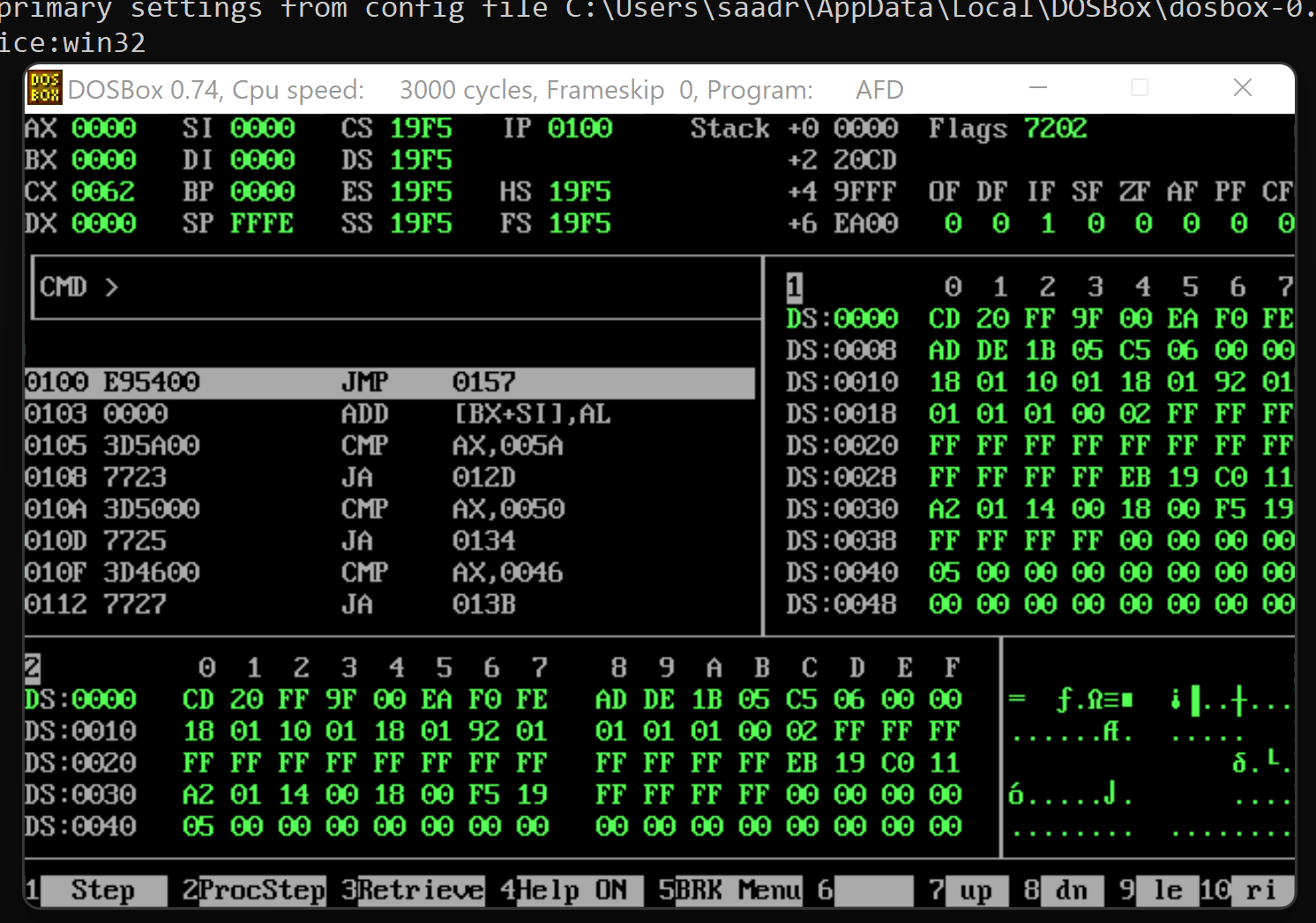
start:

mov ax,33 ;mov 33 into ax as 33 is ascii code

call displayGrade

mov ax, 0x4c00 ;termination

int 0x21



A screenshot of a computer

Description automatically generated with medium confidence

**Qno.2**

Write a subroutine to print the factorial of a number by defining a subroutine named &#39;Factorial&#39;.

Factorial of any number n is represented by n! and is equal to 1\*2\*3\*.... \*(n-1) \*n. E.g.-

4! = 1\*2\*3\*4 = 24

3! = 3\*2\*1 = 6

2! = 2\*1 = 2

**Answer**

org [0x0100]

jmp start

num: dw 0

fac: dw 2

start: ;start line

mov ax, 4

mov cx, ax

cmp ax,0

je default

dec cx

jmp factorial ;jump to label of factorial

default:

mov ax,1

mov ax,0x4c00 ;termination

factorial:

mul cx ;multiply the value of number by decrement 1 as in next line

dec cx

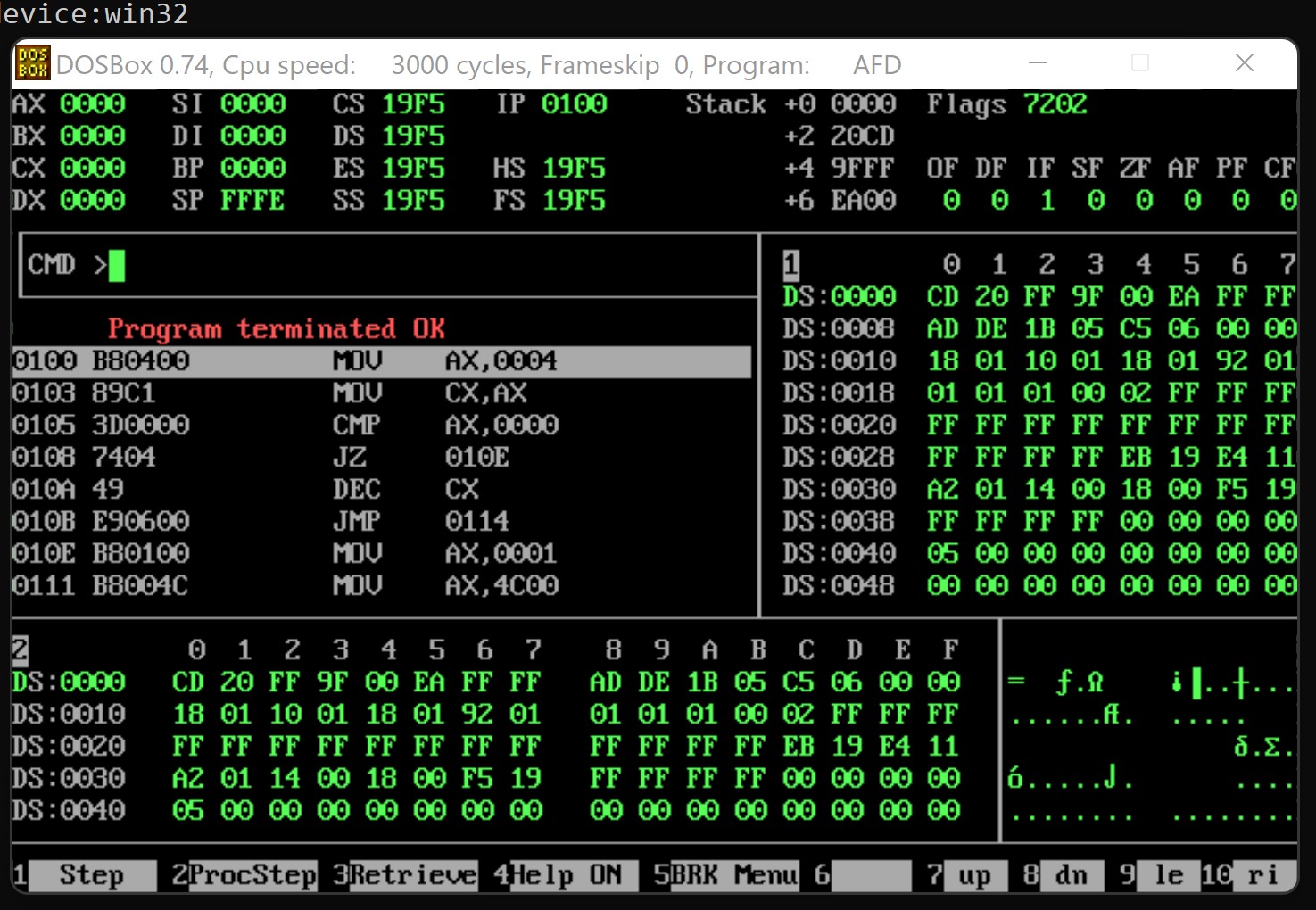
cmp cx,0 ;compare the value if after decrement 0 is reaches or not

jne factorial

mov dx,ax

mov ax,0x4c00 ;final termination

int 0x21



**Qno.3**

Implement the following two subroutines in your program. Assign proper values to registers

and variables to form the Boolean expression true or false. Call these subroutines in your main

function. Assign the subroutine name according to your will

(a): First subroutine:

if bx &gt; cx then

X = 1

(b): Second subroutine

if dx = cx then

X = 1

else

X = 2

**Answer**

[org 0x100]

jmp start

num1: dw 4,2,5,6

result: dw 0

result2: dw 0

subroutine1: ;First subroutine

mov bx,[num1+1] ;offsetadressing mode

sub ax,bx ;sub bx from ax for latest comparison

mov bx,[num1+2]

mov [result],ax

xor ax,bx ;getting both ax and bx values to zero

mov ax,[num1+3]

mov bx,[num1+4]

inc cx

cmp bx, cx ;compare bx and cx as for Question requirment

jbe subroutine1 ;jump to subroutine1 if bx is less than or equal to cx

subroutine2: ;Subroutine2

xor ax,bx

mov ax,dx

mov cx,bx

inc cx

cmp dx, cx ;Comparing dx and cx if equal then move on

jbe subroutine2 ;if equal then move on, else go to subroutine2

start:

mov cx, 4 ; bit count

mov ax,bx

call subroutine1

mov cx,4

mov ax,bx

call subroutine2

mov ax, 0x4c00 ;termination

int 0x21

Graphical user interface

Description automatically generated

**Qno.4**

Write a two-subroutine program, first subroutine program that takes 30 numbers in an

array, adds all the numbers and store in a variable and return it. Then the second subroutine

program should allow you to split the array in two parts 15,15 numbers in two arrays after that

subtracts sum of the two arrays and finally return this number and add this sum with the sum

of the 30 number array get from the previous subroutine in the main program.

**Answer**

[org 0x100]

jmp start

num1: dw 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30

result1: dw 0

result2: dw 0

subroutine1:

push bp

mov bp, sp

push es

push ax

push bx

push cx

push dx

push di

mov ax, 0xb800

mov es, ax

mov si, 0 ; initialize array index to zero

innerloop:

mov ax, [bx+si] ; load number in ax

cmp ax, [bx+si+2] ;point es to video base

mov ax, [bp+4] ;load number in ax

mov bx, 10 ;use base 10 for division

mov cx, 0

add ax,[num1] ;adding all arrays number 1 by 1

dec cx ;decrement cx as a counter

cmp si, cx ;comparing if we are at last index of array

jne innerloop ;if not equal then jump to subroutine1

mov [result1], ax

subroutine2:

mov ax, 0xb800

mov es, ax

mov si, 0 ; initialize array index to zero

innerloop1:

mov ax,bx

shl ax,1

mov cx,30

add ax,[num1]

dec cx

cmp si,cx

jne innerloop

mov [result2],ax

push ax

mov cx,30

pop di

pop dx

pop cx

pop bx

pop ax

pop es

pop bp

ret 2

start:

mov ax,4529

call subroutine1

mov cx,30

add si,0

call subroutine2

mov ax,0x4c00 ;termination

int 0x21

A screenshot of a computer

Description automatically generated with medium confidence

**The END**