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: 11  
……………………………………………………………………………**

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

**Qno.1**

You have to complete Home Task of popping the food in the Snake Game.

**Answer**

[org 0x0100]

jmp start

count: dd 0

;code to clear the screen

clearscreen:

push es

push ax

push di

push cx

mov ax,0xb800 ;video memory address

mov es,ax

mov ax,0x1720 ;color code and space ASCII

mov di,0

nextchar:

mov [es:di],ax

add di,2

cmp di,4000

jne nextchar

;popping all values

pop cx

pop di

pop ax

pop es

ret

randomNumber:

push ax;

push dx;

mov ah, 00h ; interrupts to get system time

int 1AH ; CX:DX now hold number of clock ticks since midnight

mov ax, dx

xor dx, dx

mov cx, [bp-2] ;upper number

div cx ; here dx contains the remainder of the division - from 0 to 9

add dx, 0 ; to ascii from '0' to '9'

mov bx,dx;

pop dx;

pop ax;

ret;

delay: ;a large loop for delaying next move

mov dword[count],500000

delayLoop:

dec dword[count];

cmp dword[count],0;

jne delayLoop;

ret;

printSnakePos:

push es;

push ax;

push cx;

mov ax,0x872A; ;changing the colour of the snake..and for blurring

nextloc:

mov word [es:di],ax

add di,2

dec cx

cmp cx,2

jne face ; to make snake face

mov al,0x01

face:

cmp cx,0

jne nextloc;

pop cx;

pop ax;

pop es;

ret;

printFood:

push ax;

push bx;

push di;

push es;

mov bx,80; x pos

mov [bp-2],bx;

call randomNumber;

mov [bp-4],bx;

mov bx,25;

mov [bp-2],bx;

call randomNumber;

mov [bp-2],bx ;y pos

call calculatePosition;

mov ax,0x872B ;4201

mov word [es:di],ax;

pop es

pop di

pop bx

pop ax

ret

calculatePosition:

push cx

push dx

mov cx,[bp-2] ;y pos

mov ax,80

mul cx

add ax,[bp-4] ;x pos

shl ax,1

shl ax,1

mov di,ax

mov ax,0xb800

mov es,ax

pop dx;

pop cx;

ret;

printsnake:

;pushing all values

push bp

mov bp ,sp

sub sp,4;

push di

push es

push ax

push bx

push dx

;calculating positon

mov ax,[bp+4]; y pos

mov [bp-2],ax;

mov ax,[bp+6]; x pos

mov [bp-4],ax;

call calculatePosition

mov ax,0x0702

mov dx,0

moving: ;for next move

mov cx,14; snake length

call clearscreen ;previous output clearing

call printSnakePos;

call delay;

call printFood;

call delay

inc dx;

cmp dx,30 ; total no of moves

jne moving;

;call printFood;

pop dx

pop bx

pop ax

pop es

pop di

mov sp, bp

pop bp

ret 4

start:

call clearscreen

; location of snake at start

push 4 ; row

push 4 ;col

call printsnake

mov ax,0x4c00

int 0x21



**Q no. 2:**

Str\_concat ProcedureWrite a procedure named Str\_concat that concatenates a source string to the end of a target string. Sufficient space must exist in the target string to accommodate the new characters. Pass pointers to the source and target strings.

**Answer**

[ORG 0x100]

jmp start

message: db 'SAAD'

result: db 'REHMAN '

lenght: dw 10

clrscn:

push es

push ax

push di

mov ax , 0xb800

mov es,ax

mov di , 0

nextpos:

mov word[es:di] , 0x0720

add di, 2

cmp di , 4000

jne nextpos

pop di

pop ax

pop es

ret

concatstr:

push bp

mov bp, sp

push es

push ax

push cx

push si

push di

mov di, [bp+8]

add di, 5

mov si, [bp+6]

mov cx, [bp+4]

sub cx , 5

mov ah, 0x07

conchar:

mov al, [si]

mov [di], al

add di, 1

add si, 1

loop conchar

pop di

pop si

pop cx

pop ax

pop es

pop bp

ret 6

printstr:

push bp

mov bp, sp

push es

push ax

push cx

push si

push di

mov ax, 0xb800

mov es, ax

mov di, 160

mov si, [bp+6]

mov cx, [bp+4]

mov ah, 0x07

nextchar:

mov al, [si]

mov [es:di], ax

add di, 2

add si, 1

loop nextchar

pop di

pop si

pop cx

pop ax

pop es

pop bp

ret 4

start:

call clrscn

mov ax , result

push ax

mov ax , message

push ax

push word[lenght]

call concatstr

mov ax , result

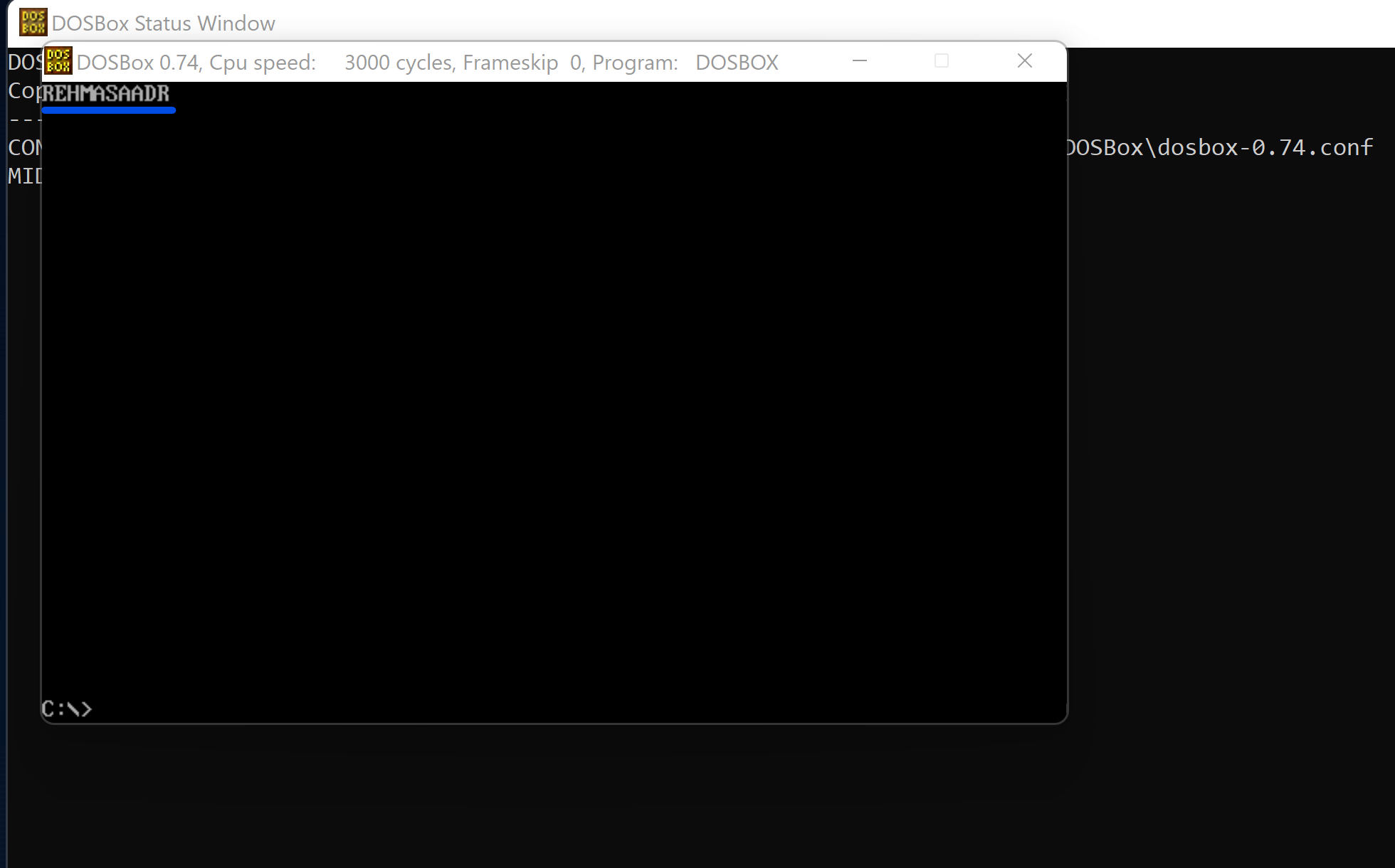
push ax

push word[lenght]

call printstr

mov ax, 0x4c00

int 0x21



**Q no. 3:**

Str\_remove ProcedureWrite a procedure named Str\_remove that removes n characters from a string. Pass a pointer to the position in the string where the characters are to be removed. Pass an integer specifying the number of characters to remove.

**Answer**

[org 0x0100]

jmp start

source: db 'My name is SAAD'

length1: dw 15

clear\_scr:

push ax

push es

push di

mov ax, 0xb800

mov es, ax

mov di, 0

clr:

mov word [es:di], 0x0720

add di, 2

cmp di, 4000

jne clr

pop di

pop es

pop ax

ret

remove:

push bp

mov bp, sp

push ax

push bx

push cx

push si

push di

mov bx, [bp+8] ; reduces the size length

mov cx, [bx]

sub cx, [bp+6]

sub cx, [bp+4]

mov ax, [bp+4]

sub [bx], ax

mov si, [bp+10]

add si, [bp+6]

mov di, si

add di, [bp+4]

del:

mov al, [di]

mov [si], al

inc si

inc di

loop del

pop di

pop si

pop cx

pop bx

pop ax

pop bp

ret 8

printstr:

push bp

mov bp, sp

push es

push ax

push cx

push si

push di

mov ax, 0xb800

mov es, ax ; point es to video base

mov al, 80 ; load al with columns per row

mul byte [bp+10] ; multiply with y position

add ax, [bp+12] ; add x position

shl ax, 1 ; turn into byte offset

mov di,ax ; point di to required location

mov si, [bp+6] ; point si to string

mov cx, [bp+4] ; load length of string in cx

mov ah, [bp+8] ; load attribute in ah

cld ; auto increment mode

nextchar:

lodsb ; load next char in al

stosw ; print char/attribute pair

loop nextchar ; repeat for the whole string

pop di

pop si

pop cx

pop ax

pop es

pop bp

ret 10

start:

call clear\_scr ; call the clrscr subroutine

push source

push length1

push 5

push 4

call remove

mov ax, 30

push ax ; push x position

mov ax, 20

push ax ; push y position

mov ax, 7 ; blue on black attribute

push ax ; push attribute

mov ax, source

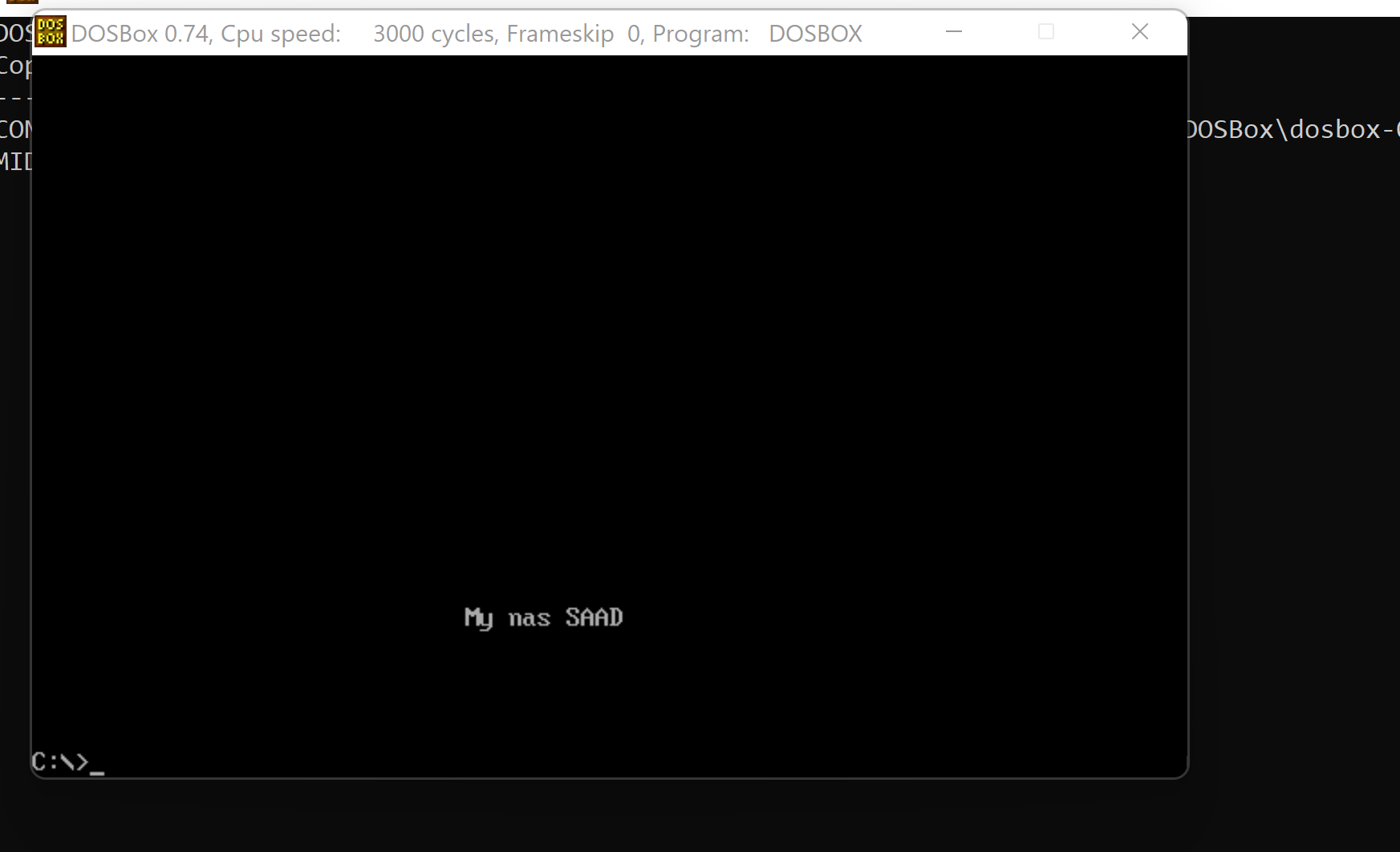
push ax ; push address of message

push word[length1] ; push message length

call printstr ; call the printstr subroutine

mov ax, 0x4c00 ; terminate program

int 0x21



**The End**