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: 03  
……………………………………………………………………………**

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

**Qno.1**

Write the assembly program to add 5, 9 and 14. Store them in variables and use direct

Addressing mode. All the numbers are word type (16 bits).

**Answer**

[ORG 0x100]

mov ax,[num1]

add ax,[num2]

add ax,[num3]

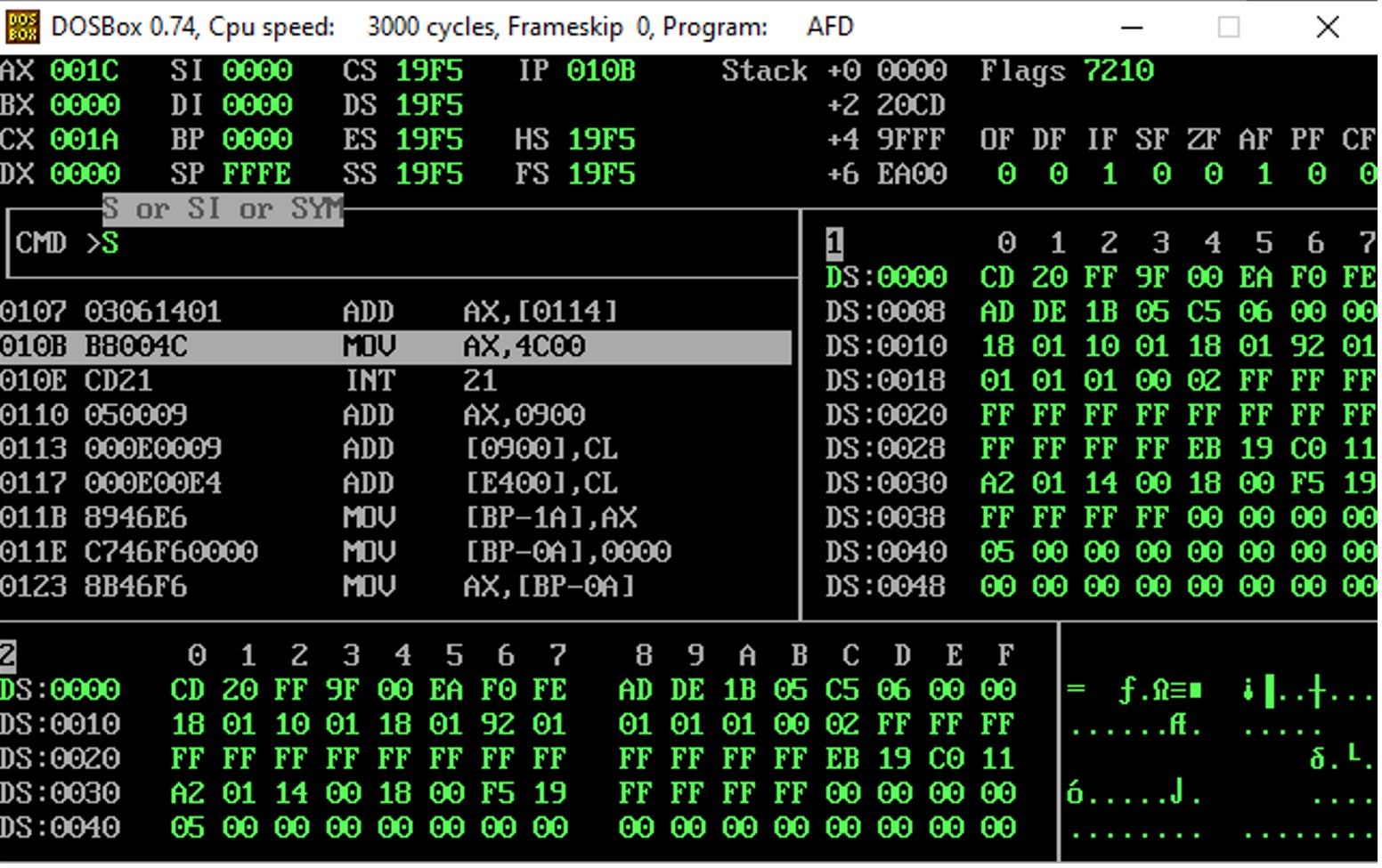
mov ax,0x4c00

int 0x21

num1: dw 5

num2: dw 9

num3: dw 1



**Qno.2**

Write the assembly program to subtract 18 from 30. Store them in Appropriate variables

and use register indirect Addressing mode. All the numbers are Byte type (Byte bits).

**Answer**

[ORG 0x100]

mov bx ,num1

mov ax,[num1]

sub ax, [num1+2]

mov ax,0x4c00

int 0x21

num1: dw 30,18

A screenshot of a computer

Description automatically generated with medium confidence

**Qno.3**

Write the following program in assembly to Find the values of result1 and result2

variables.

x = 5

y = 10

result1 = x + y – 3

result2 = result1 - 9 + y

Store x and y as memory variables and then store result in result1 and result2 variables

Using register indirect addressing mode. All the numbers are word type (16 bits)

**Answer**

[ORG 0x100]

mov bx,x

mov ax,[y]

add ax,[bx]

sub ax,3

mov [result1],ax

sub ax,9

mov bx,y

add ax,[bx]

mov [result2],ax

mov ax,0x4c00

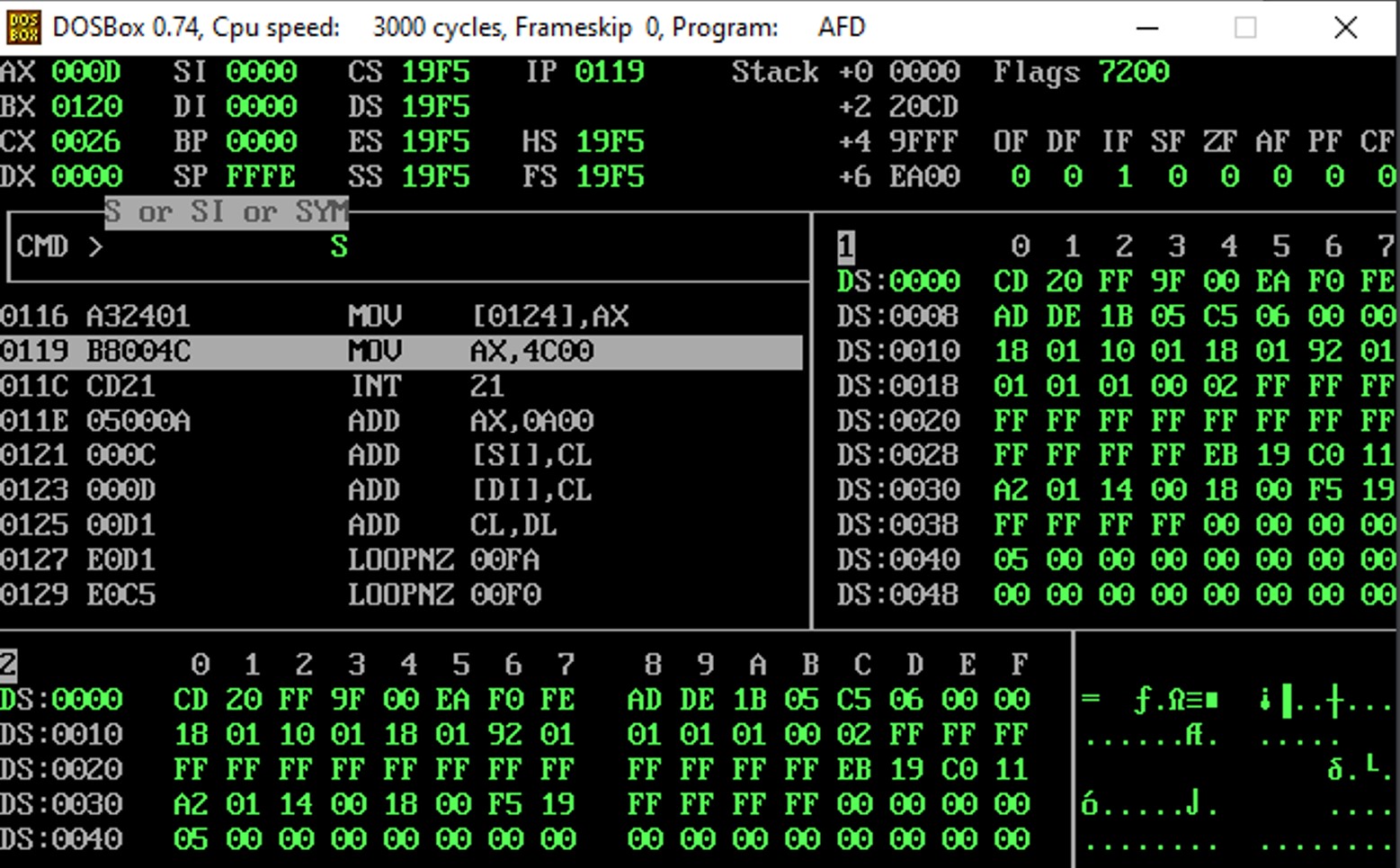
int 0x21

x: dw 5

y: dw 10

result1: dw 0

result2: dw 0



**Qno.4**

Write the assembly program to add all numbers in the range (20 - 43). Save numbers in

num1 variable. Then add all the numbers and save the result in the sum variable. Use

register + offset addressing mode in this question. num1 label will have following

declarations, you have to implement them one by one:

a. Define Byte (8 bits)

b. Define word (16 bits)

**Answer**

**(b)**

[org 0x0100]

mov bx, 0

mov cx, 23

mov ax, 0

l1: add ax, [num1+bx]

add bx, 2

sub cx, 1

jnz l1

mov [total], ax

mov ax, 0x4c00

int 0x21

num1: dw 20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43

total: dw 0

A screenshot of a computer

Description automatically generated with medium confidence

**(a)**

[org 0x0100]

mov bh, 0

mov ch, 23

mov ah, 0

l1: add ah, [num1+bh]

add bh, 2

sub ch, 1

jnz l1

mov [total], ah

mov ax, 0x4c00

int 0x21

num1: dw 20,21,22,23,24,25,26,27,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43

total: dw 0

A screenshot of a computer

Description automatically generated with medium confidence

**Qno.5**

Take a number in a 16-bit variable from 0 – 9 and then take another number in another

variable. Take the value of your choice. Then perform the following operations using

indirect addressing:

a. Increment the num1 variable.

b. Decrement the num2 variable.

c. Swap the num1 and num2 variables.

d. Display the value of num1 variable on AFD.

e. Display the value of num2 variable on AFD.

**Answer**

[org 0x0100]

mov bx, var1

mov ax, [bx]

add ax, 1

mov bx, var2

mov cx, [bx]

sub cx, 1

xor ax, cx

xor cx, ax

xor ax, cx

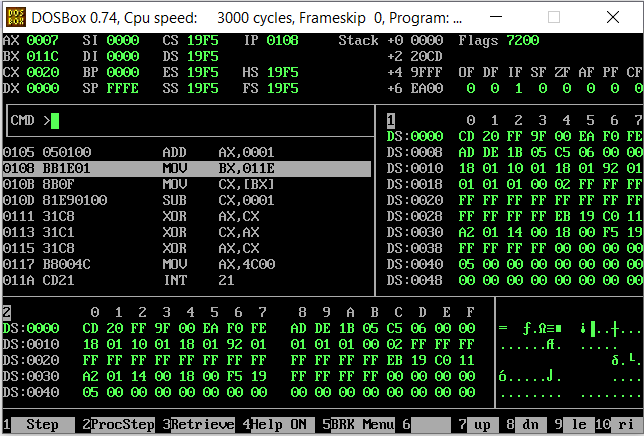
mov ax, 0x4c00

int 0x21

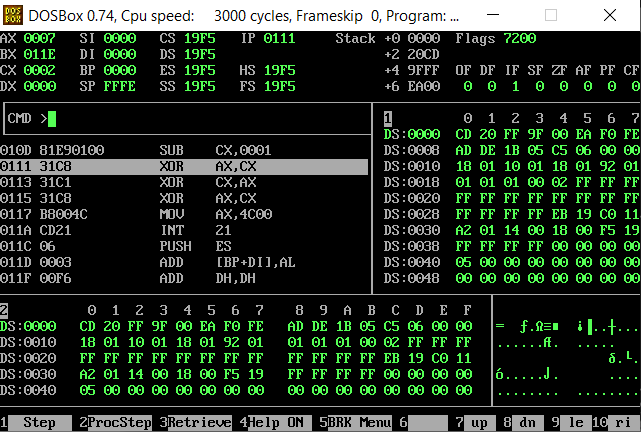
var1: dw 6

var2: dw 3

**(a)**



**(b)**



**(c)**

A screenshot of a computer

Description automatically generated with medium confidence

**Qno.6**

Create an array of any size and give some values to it. Then count all the elements of the

array and store count in a memory variable. Also Display count on AFD appropriately.

**Answer**

[org 0x0100]

mov bx, var1

mov cx, [var2]

mov ax, [var3]

l1:

add ax, 1

add bx, 2

sub cx, 3

jnz l1

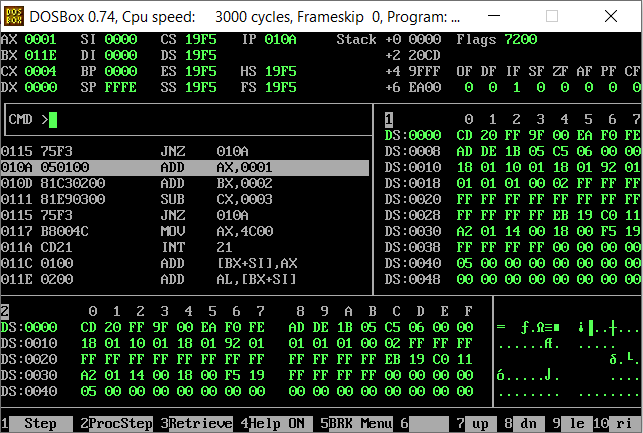
mov ax, 0x4c00

int 0x21

var1: dw 1,2,3,4,5,6,7

var2: dw 7

var3: dw 0



**The End**