|  |
| --- |
| FAST National University |
| **Introduction to DOSBox**  **Lab 1** |
|  |

**Computer Organization and Assembly Language**

|  |  |
| --- | --- |
| **Student Name** | Umamah Hussain |
| **Roll Number** | 21L-1858 |
| **Instructor** | Hazoor Ahmad |
| **Class** | CS3 |
| **Section** | A1, A2, D1, D2 |
| **Semester** | Fall 2022 |

Fast School of Computing

FAST-NU, Lahore, Pakistan

# Activity 1

|  |  |  |  |
| --- | --- | --- | --- |
| Zero Flag | Carry Flag | Sum Flag | Overflow Flag |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |

# Activity 2: square

## **Assembly Language Code**

[org 0x0100]

mov cx, [num1]

mov ax, 0

mov bx, num1

l1: add ax, [bx]

sub cx, 1

jnz l1

mov [square], ax

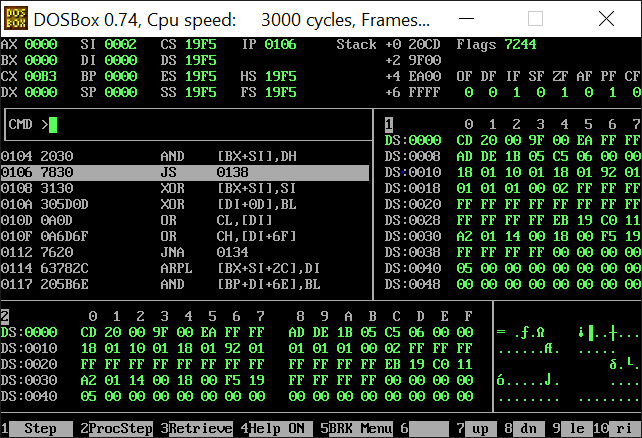
mov ax, 0x4c00

int 0x21

num1: dw 5

square: dw 0

## **Debugging Screenshots**



# Activity 4:factorial

## **Assembly Language Code**

[org 0x0100]

jmp start

num: dw 6

fac: dw 0

start:

mov ax, [num]

mov dx, ax

mov cx, ax

sub cx, 1

mov bx, 1

first:

add [fac], dx

sub cx, 1

jnz first

sub ax, 1

l1:

sub ax, 1

cmp ax, bx

jne calcfac

jmp exit

calcfac:

mov cx, ax

sub cx, 1

mov dx, [fac]

l2:

add [fac], dx

sub cx, 1

jnz l2

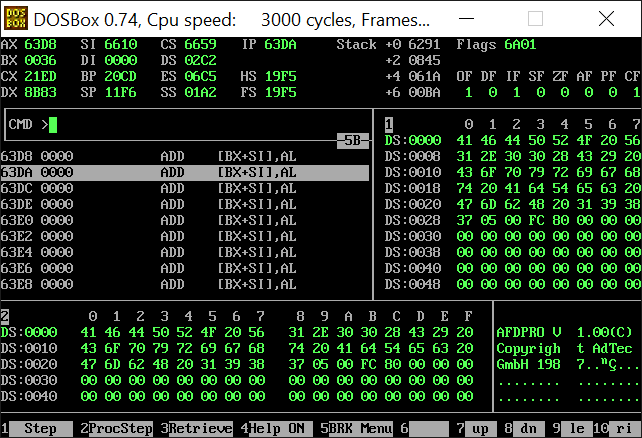
jmp l1

exit:

mov ax, 0x4c00

int 0x21

## **Debugging Screenshots**



# Activity 3: frequency

## **Assembly Language Code**

[org 0x0100]

jmp start

arr: dw 2,3,4,7,6,2,8,9,4,2

num: dw 2

start:

mov ax, 0

mov cx, 10

mov si, 0

mov dx, 0

l1:

mov ax, [arr+si]

add si, 2

cmp ax, [num]

je addfreq

sub cx, 1

jnz l1

jmp exit

addfreq:

add dx, 1

sub cx, 1

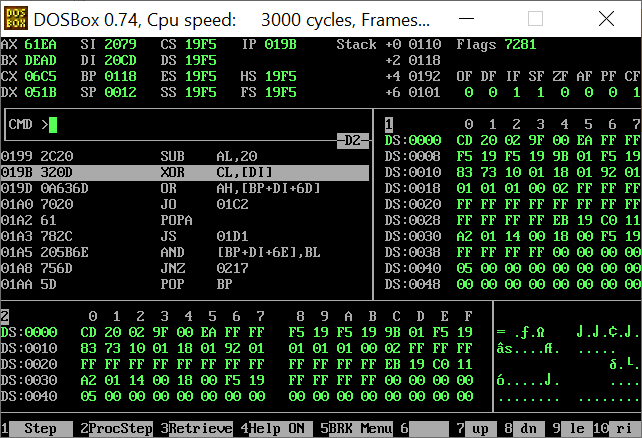
jnz l1

exit:

mov ax, 0x4c00

int 0x21

## **Debugging Screenshots**



# Activity 5:max

## **Assembly Language Code**

[org 0x0100]

mov bx,0

l1:mov ax,[array+bx]

l2:add bx,2

cmp ax,[array+bx]

cmp bx,24

je l3

ja l2

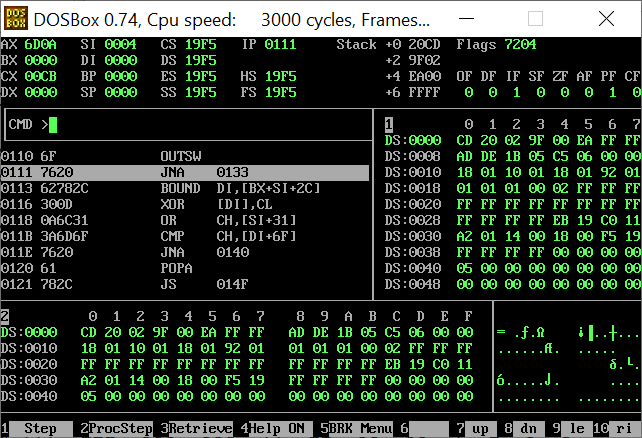
jbe l1

l3:mov ax, 0x4c00

int 0x21

array: dw 111, 999, 888, 888, 11, 99, 88, 88, 1, 9, 8, 8

## **Debugging Screenshots**



# Activity 6

## **Assembly Language Code**

[org 0x0100]

mov ax,[array]

mov bx,[array]

mov si,0

mov cx,[maxsize]

Max:

cmp ax,[array+si]

jnl Secondmax

mov ax,[array+si]

Secondmax:

cmp bx,[array+si]

jnl looop

cmp ax,[array+si]

je looop

mov bx,[array+si]

looop:

add si, 2

sub cx,1

jnz Max

mov word[max],ax

mov word[secondmax],bx

mov ax, 0x4c00 ;terminate quit the program

int 0x21

array: dw 111, 999, 888, 888, 11, 99, 88, 88, 1, 9, 8, 8

maxsize: dw 12

max: dw 0

secondmax: dw 0

## **Debugging Screenshots**

