NATIONAL UNIVERSITY
of Computer & Emerging Sciences

Coal

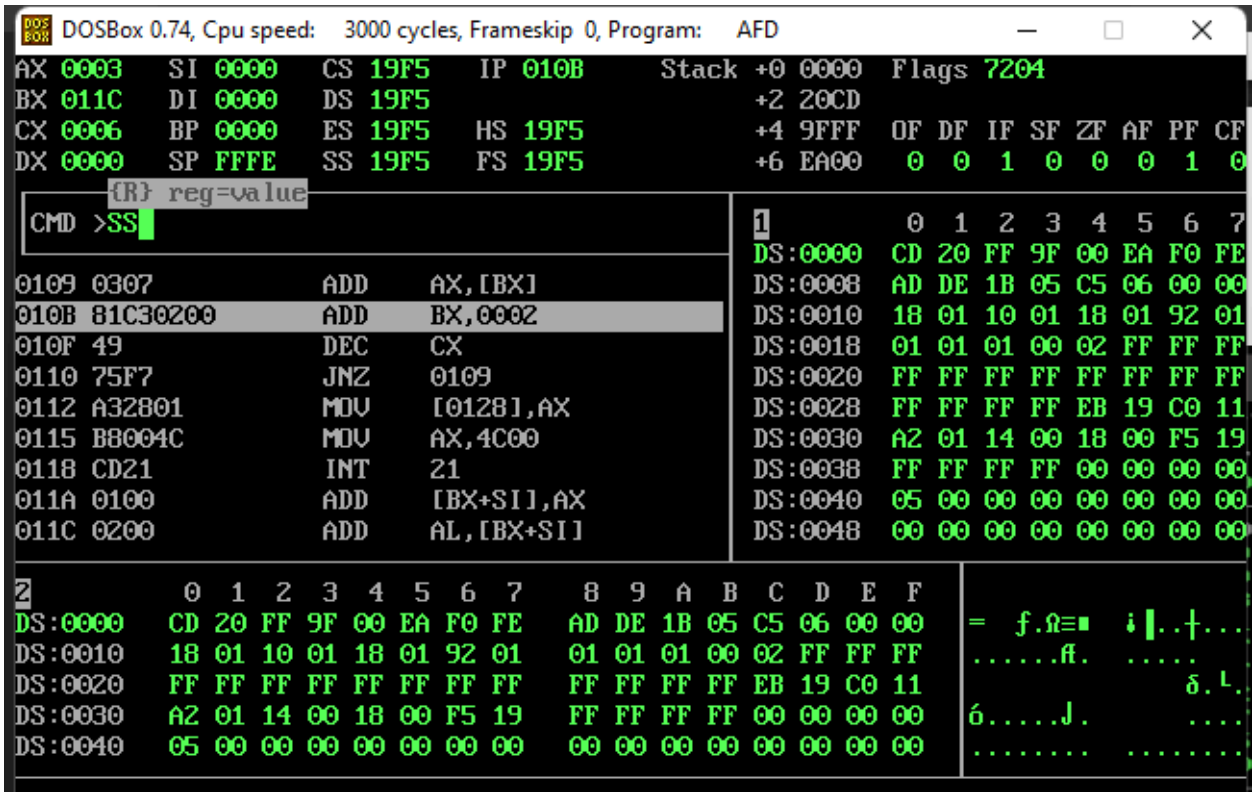
LAB TASK 5

NAME SAJID ISLAM ROLLNO 24p-0745 INSTRUCTOR Muhammad Hassan

Question 1

```
[org 0x0100]

mov cx, 7         ; loop counter
mov ax, 0         ; clear accumulator
mov bx, numbers   ; point to numbers array

loop_start:
add ax, [bx]      ; add value to ax
add bx, 2         ; move to next word
dec cx            ; decrement counter
jnz loop_start    ; loop until cx=0

mov [result], ax ; store result

mov ax, 0x4c00    ; exit program
int 0x21

numbers: dw 1,2,3,4,5,6,7
result:  dw 0
```

Output



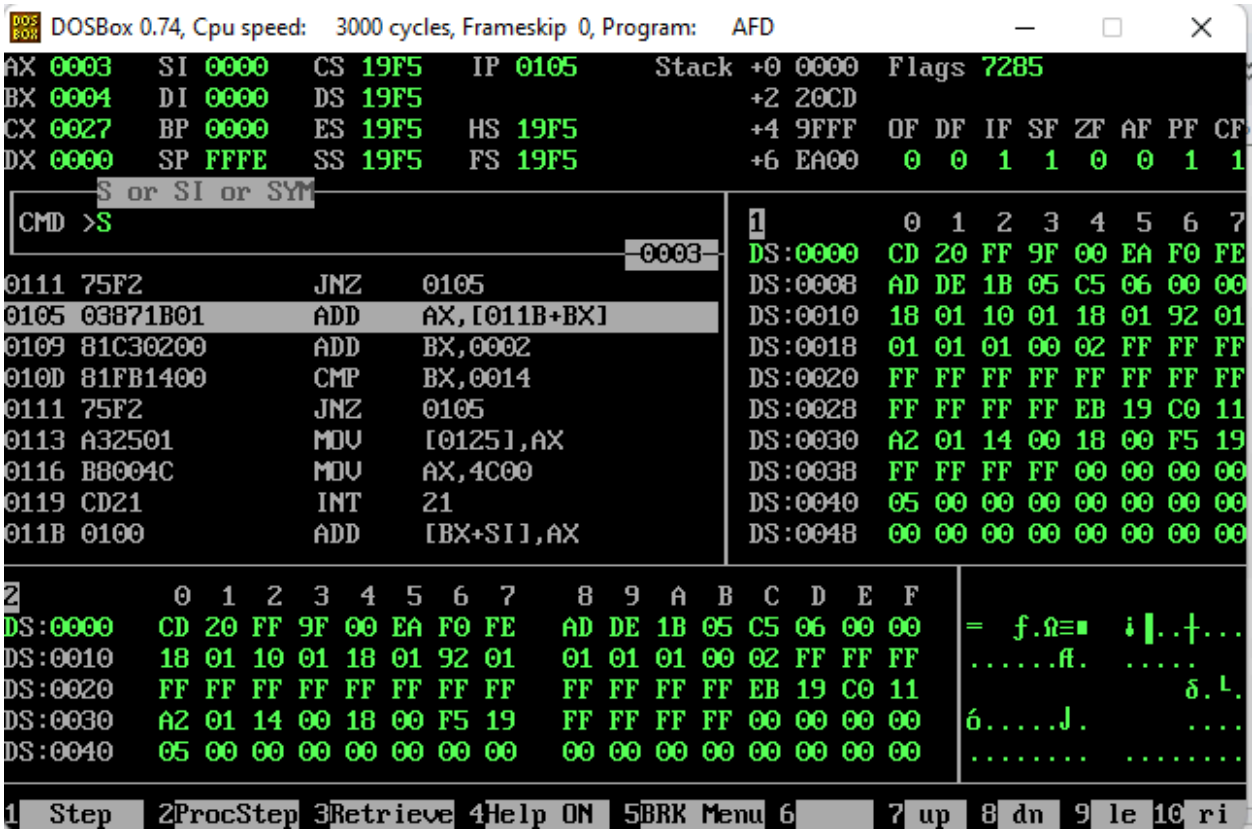
Question 2

```
[org 0x0100]

xor ax, ax        ; clear ax
mov bx, 0         ; index

loop_start:
add ax, [data + bx]  ; add value at data+bx to ax
add bx, 2            ; increment index
cmp bx, 20           ; check if end of array
jne loop_start       ; loop if not done

mov [result], ax     ; store result

mov ax, 0x4c00       ; exit program
int 0x21

data:   dw 1,2,3,4,5
result: dw 0
```

used CMP BX, 20 and JNE to stop when BX reaches the end of the array.

Output

Screen Shot after first iteration

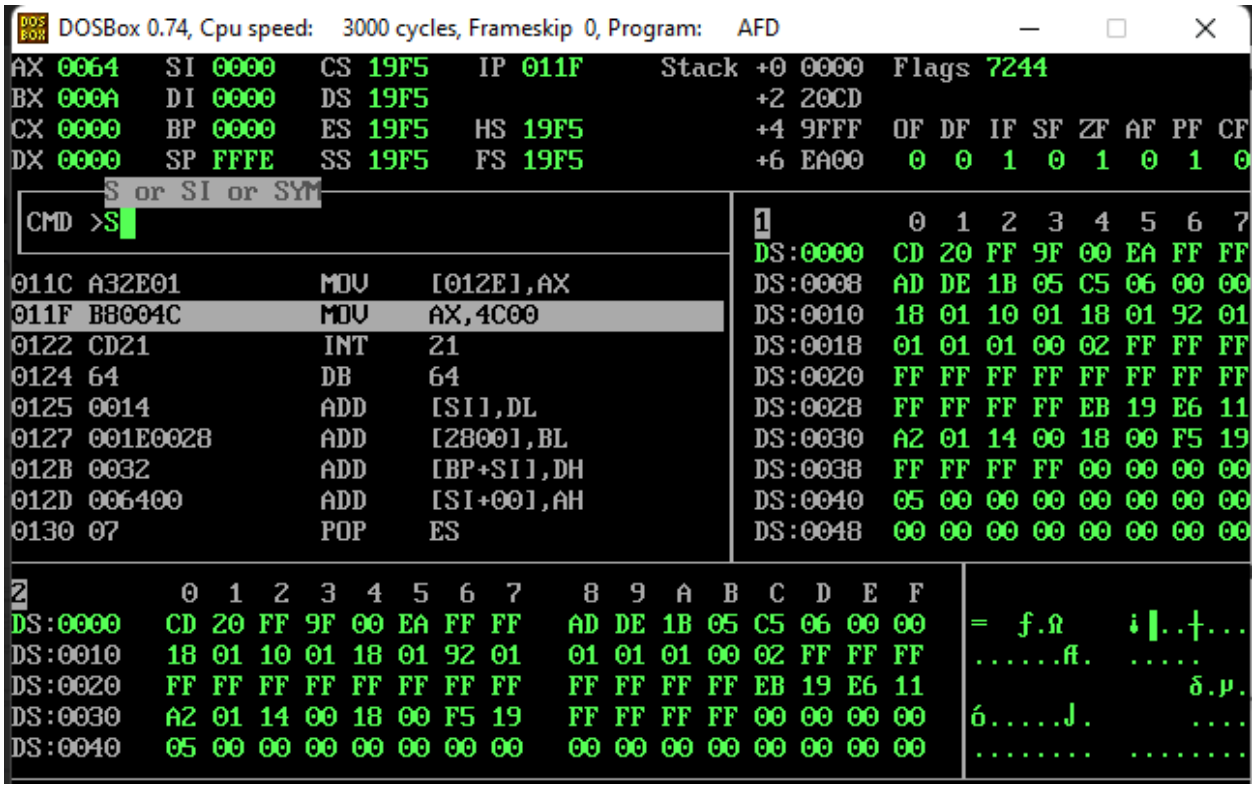


Question 3

```
[org 0x0100]

mov ax, [array]  ; moving element at first index into ax
mov bx, 2

loop_start:
cmp ax, [array + bx]  ; compare ax with current element
jg no_swap            ; if ax > [array+bx], skip
jl swap               ; if ax < [array+bx], update max

swap:
mov ax, [array + bx]  ; update ax with new max

no_swap:
add bx, 2             ; move to next word
cmp bx, 10
jne loop_start        ; loop if not done

mov [maximum], ax     ; store maximum value

mov ax, 0x4c00
int 0x21

array:   dw 100, 20, 30, 40, 50
maximum: dw 0
```

Output

Screenshot after storing the largest number in memory



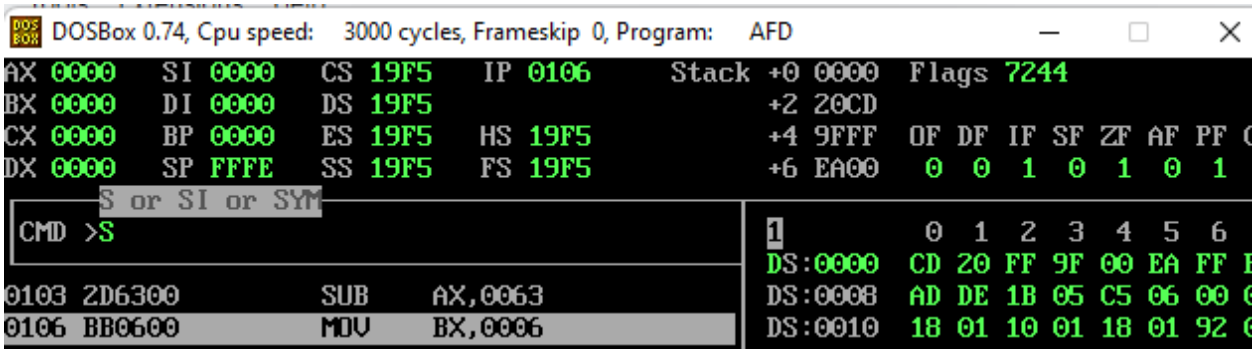
Question 4

Zero Flag

```
[org 0x0100]

mov ax, 99
sub ax, 99          ; ZF = 1

mov bx, 6
sub bx, 5           ; ZF = 0

mov ax, 0x4c00
int 0x21
```

Output

After executing the subtraction command we get the below results

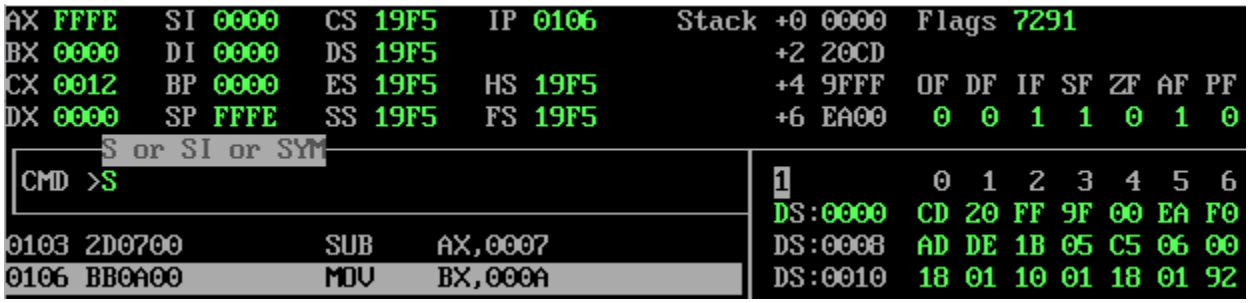


Zero flag becomes zero again after executing the end command

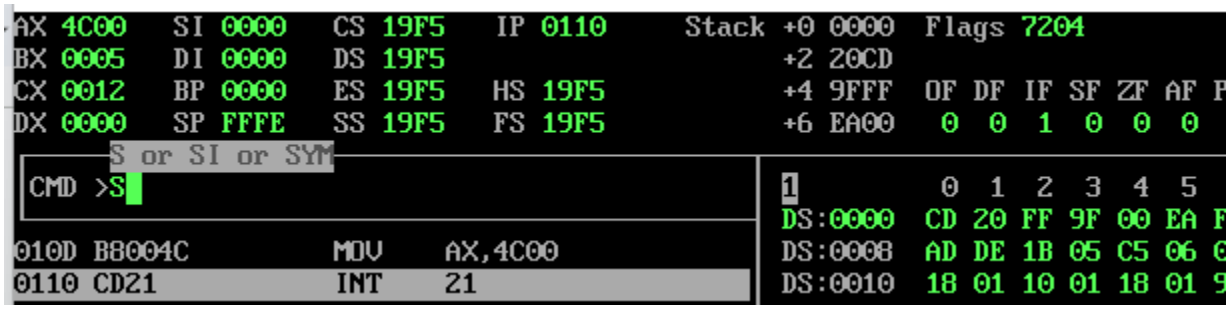
Sign flag

```
[org 0x0100]

mov ax, 5
sub ax, 7          ; SF = 1

mov bx, 10
sub bx, 5          ; SF = 0

mov ax, 0x4c00
int 0x21
```

Output



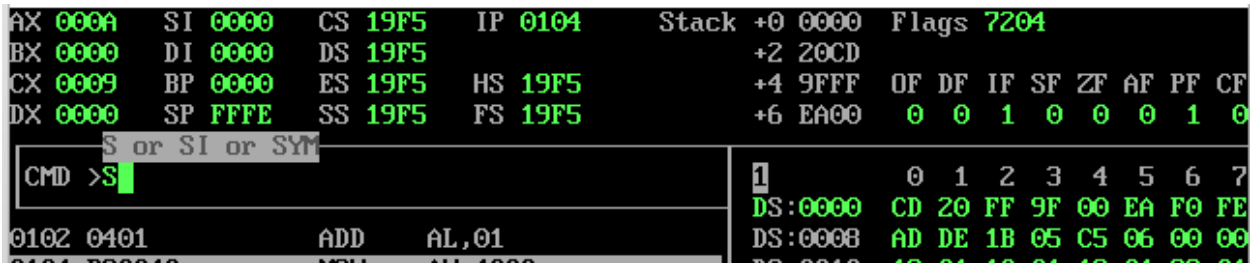
The sign flag again becomes zero at end command



Parity Flag

```
[org 0x0100]

mov al, 9
add al, 1          ; PF = 1 (even parity)

mov ax, 0x4c00
int 0x21
```

Output

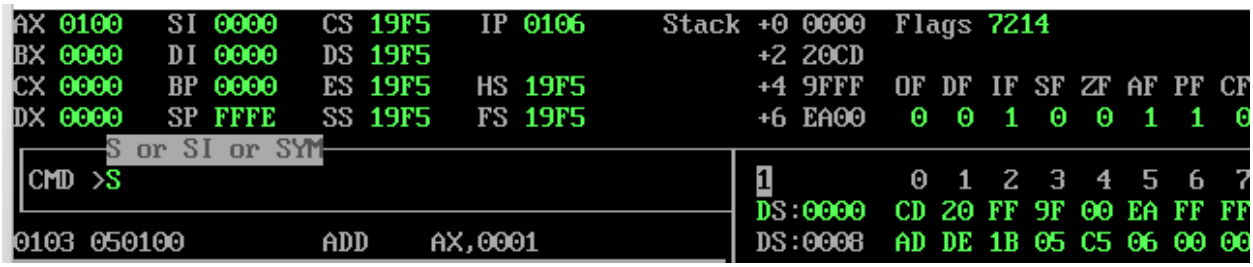


Auxiliary Flag

```
[org 0x0100]

mov ax, 255
add ax, 1          ; AF = 1 (carry from lower nibble)

mov bx, 0x05
add bx, 1          ; AF = 0

mov ax, 0x4c00
int 0x21
```

Output

Auxilary flag become on



Auxiliary flag become zero again

Question 5

```
[org 0x0100]              ; Origin address

mov bx, data              ; BX points to start of data array
mov dx, result            ; DX points to start of result array
mov cx, 6                 ; We have 6 numbers to process

next_num:
    mov ax, [bx]          ;
    cmp ax, 0             ; Compare with 0
    jne non_zero          ; Jump if not equal to 0

; if the number is zero
    mov byte [dx], 0      ;
    jmp store_done

non_zero:
    mov byte [dx], 1      ; Store 1 in result array

store_done:
    add bx, 2             ; Go to next number (word = 2 bytes)
    inc dx                ; Next result (byte)
    loop next_num         ; Repeat until CX=0

mov ax, 0x4c00            ; DOS terminate
int 0x21

data:   dw 0, -5, 12, 7, -128, 255
result: db 0, 0, 0, 0, 0, 0
```

Output

This is the final output