COAL LAB - REPORT 5

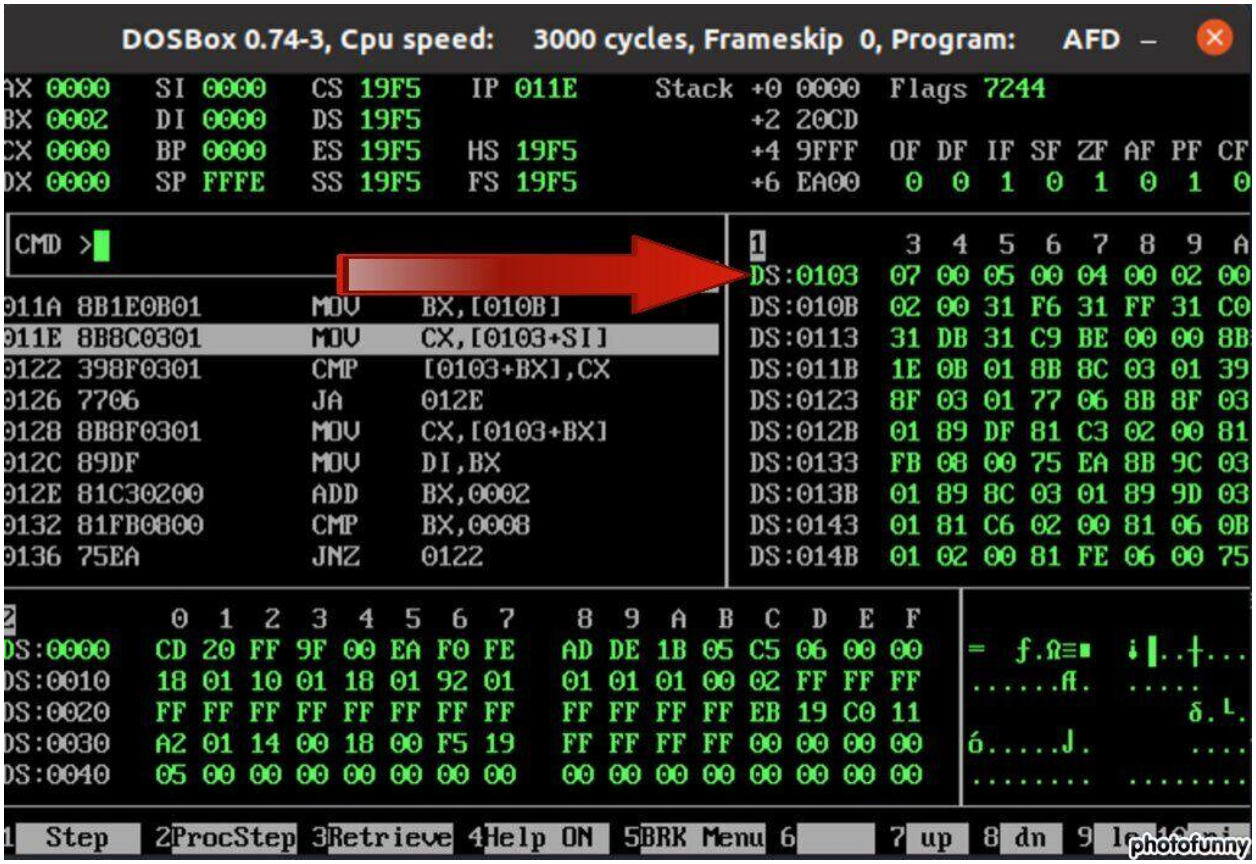
Content covered

- ❖ Sorting
- ❖ Nested loops
- ❖ System Delays

TASK - 1

SELECTION SORT

Selection sort is a simple sorting algorithm that finds the minimum number in an array and swaps it with the leftmost element. This swapped element becomes part of the sorted array and for the next iteration it repeats the same process i.e: finds minimum element and swaps it with the leftmost element of unsorted array. These steps are repeated until all elements of array are traversed.

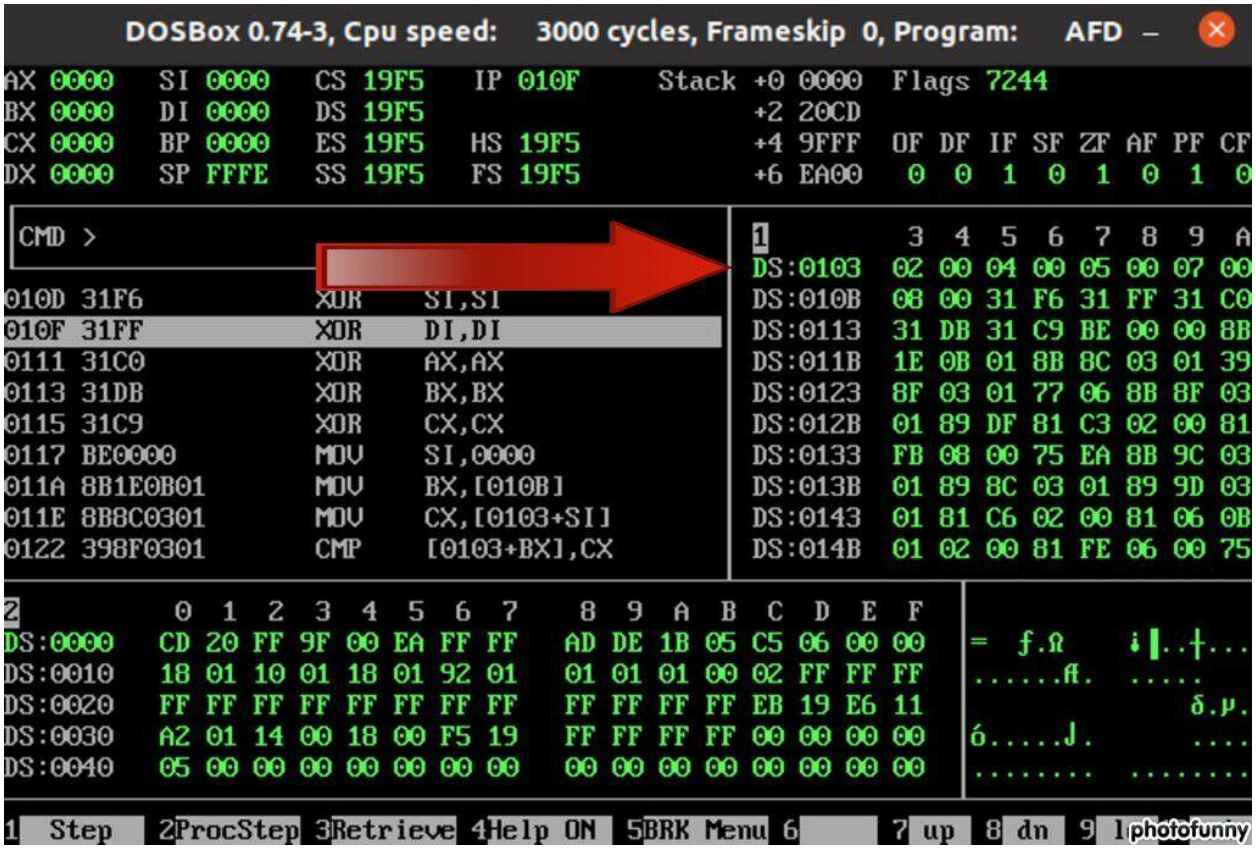


Here you can see 7, 5, 4, 2 are in this address after the execution of selection sort it will be converted into 2, 4, 5, 7 as shown in below pic

```
[org 0x0100]

jmp start

data : dw 7, 5, 4 , 2

counter: dw 2

start:
        ;Making all registers 0

        xor si, si
        xor di, di
        xor ax, ax
        xor bx, bx
        xor cx, cx
```

```
        mov si, 0  ;

        outerloop:

                mov bx , [counter]

; Moving first number in bx with the help of counter label and this counter will be incremented below

                mov cx , [data + si] ;


                Sort:
                        cmp [data+bx], cx

                        ja noswap

; here we are comparing the 5 with 7 and if 5 will be greater then there will be no swap  other wise 5 will be stored in cx register as a minimum for at the moment

                        mov cx, [data+bx]

                        mov di, bx

 ; And in di we are moving the index of that  number which is  minimum because at the end we have to shift the minimum with first number

                noswap:

                        add bx, 2
                        cmp bx, 8

                        jne sort


                ; code for swapping the first number with minimun


                mov bx, [data+si]

                mov [data+si], cx

                mov [data+di], bx


                add si, 2

                add word[counter], 2

                cmp si, 6

                jne outerloop


                mov ax, 0x4c00
```

int 0x21



Now the array is sorted 2, 4, 5, 7

TASK - 2

DELAY

Actually the microprocessor is very fast so we want some delay…

```
[org 0x100]
        start:
        mov ax, 255
        againl:
        mov bx,255
        again2:
        mov dx, 10
        outerloop:
        dec ax
        jne outerloop
        innerloop:
        dec bx
        jne innerloop:


        mov ax,0x4c00
        int 0x21
```