

# COAL LAB - REPORT 5

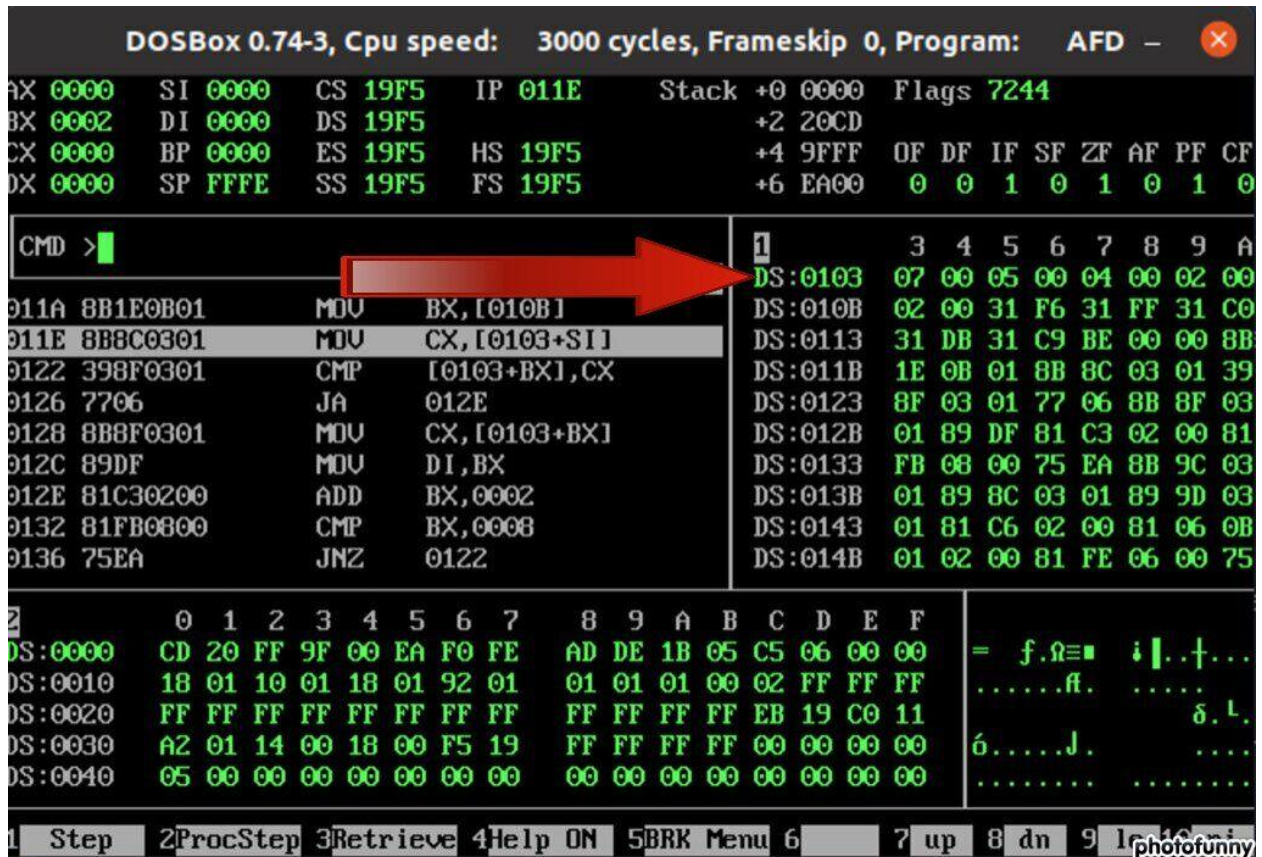
## Content covered

- ❖ 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 minimum

```
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

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: AFD -  
AX 0000 SI 0000 CS 19F5 IP 010F Stack +0 0000 Flags 7244  
BX 0000 DI 0000 DS 19F5 +2 20CD  
CX 0000 BP 0000 ES 19F5 HS 19F5 +4 9FFF OF DF IF SF ZF AF PF CF  
DX 0000 SP FFFE SS 19F5 FS 19F5 +6 EA00 0 0 1 0 1 0 1 0  
CMD >  
010D 31F6 XOR SI,SI  
010F 31FF XOR DI,DI  
0111 31C0 XOR AX,AX  
0113 31DB XOR BX,BX  
0115 31C9 XOR CX,CX  
0117 BE0000 MOV SI,0000  
011A 8B1E0B01 MOV BX,[010B]  
011E 8B8C0301 MOV CX,[0103+SI]  
0122 398F0301 CMP [0103+BX],CX  
1 3 4 5 6 7 8 9 A  
DS:0103 02 00 04 00 05 00 07 00  
DS:010B 08 00 31 F6 31 FF 31 C0  
DS:0113 31 DB 31 C9 BE 00 00 8B  
DS:011B 1E 0B 01 8B 8C 03 01 39  
DS:0123 8F 03 01 77 06 8B 8F 03  
DS:012B 01 89 DF 81 C3 02 00 81  
DS:0133 FB 08 00 75 EA 8B 9C 03  
DS:013B 01 89 8C 03 01 89 9D 03  
DS:0143 01 81 C6 02 00 81 06 0B  
DS:014B 01 02 00 81 FE 06 00 75  
2 0 1 2 3 4 5 6 7 8 9 A B C D E F  
DS:0000 CD 20 FF 9F 00 EA FF FF AD DE 1B 05 C5 06 00 00 = f. n i | . + ...  
DS:0010 18 01 10 01 18 01 92 01 01 01 01 00 02 FF FF FF .....ff. ....  
DS:0020 FF FF FF FF FF FF FF FF FF FF FF FF FF EB 19 E6 11 .....δ.μ.  
DS:0030 A2 01 14 00 18 00 F5 19 FF FF FF FF 00 00 00 00 6.....J. ....  
DS:0040 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....  
1 Step 2ProcStep 3Retrieve 4Help ON 5BRK Menu 6 7 up 8 dn 9 1photofunny
```

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
again1:
mov bx, 255
again2:
mov dx, 10
outerloop:
dec ax
jne outerloop
innerloop:
dec bx
jne innerloop:

mov ax, 0x4c00
int 0x21
```