Shoaib Akhtar

20P-0147

LAB-REPORT-8

Content of lab 8: Subroutines

1. A function is a block of code that only runs when it is called. A function can be called anywhere after the function has been declared. By itself, a function does nothing.

2. The main advantage of subroutine is reusability. For example we have to print something on dosbox screen, for that we have two options. One is that we write the code again and again for printing and the second one is that we only make a function one time and by doing this we can save a lot of time. This was the mistake I made in my project.

Here is the code of the selection sort which I did in the past. Now I have converted it to **subroutine**. Now I can use it as I want.

```
[org 0x0100]

jmp start

data : dw 7, 5, 4 , 2

counter: dw 2


selectionSort:


        xor si, si
        xor di, di
        xor ax, ax
```

```asm
        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]
```

```asm
        mov [data+si], cx

        mov [data+di], bx


        add si, 2

        add word[counter], 2

        cmp si, 6

        jne outerloop


    ret


start:

    call selectionSort

    mov ax, 0x4c00
    int 0x21
```