**Microprocessor Lab  
Lab Experiment No. 9**

Name: Ninad Rao Roll No. 53

**Aim**: Perform sorting of an integer array.

**Instructions on how to use TASM**:

Steps for creating the program:

1. TASM is loaded
2. TASM < Edit - We will get an edit window
3. Type the program here
4. Save the file as <filename>.asm

Steps for running the program:

1. c:\tasm> Type here tasm filename

**c:\tasm> tasm <filename>.asm**

This will save the program, and the edit window with this file name will be seen.

1. c:\tasm> Linking the program

**c:\tasm> tlink <filename>.obj**

This will create an object file after linking.

1. c:\tasm> Now to execute the program and get to the result window

**c:\tasm> td <filename>.exe**

After execution, all the window options are present to check all registers, all memory locations and so on.

**Program to sort an integer array**:

**Explanation**: To sort the numbers, let us assume that there are ten numbers in the array and its starting address is 01H. Initially, counter-1 and counter-2 are initialized with the size of the array. In the first iteration, the first number is compared with the second number. In the second iteration, the first number is compared with the third number. In the third iteration, the first number is compared with the fourth number and the process continues until counter-2 becomes zero and the numbers are arranged in ascending order.

**Algorithm**:

**Step I:** Load data from offset 500 to register CL (for count).

**Step II:** Travel from starting memory location to last and compare two numbers if the first number is greater than the second number then swap them.

**Step III:** First pass fix the position for the last number.

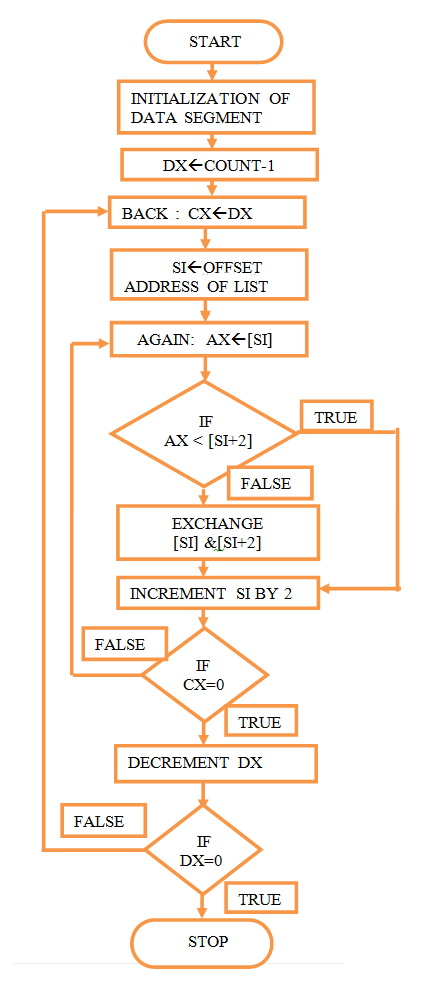
**Step IV:** Decrease the count by 1.

**Step V:** Again travel from starting memory location to (last-1, by help of count) and compare two numbers if the first number is greater than the second number then swap them.

**Step VI:** Second pass fixes the position for the last two numbers.

**Step VII:** Repeat

**Flowchart**:



**Code**:

.model small

.data

num DB 01h, 02h, 12h, 54h, 00h, 70h, 15h, 45h, 11h, 25h

count db 0Ah

.code

mov ax, @data ; Initialization of data pointer

mov ds, ax ; Loading of data segment

mov cl, count ; Count (10) in CL

mov ch, 00h ; Clear higher counter

dec cx ; Decrement cx (09 numbers to be sorted with first as reference)

outer: mov bx, cx ; Outer as label were count -1 put in BX

lea si, num ; Load pointer in SI

inner: mov al, [si] ; SI pointed value put in AL

inc si ; Next value

cmp al, [si] ; First and next value compared

jc over ; If carry the go to over for next data

mov dl, al ; Else swap the data items……data1 in DL

mov al, [si] ; Data 2 in AL

mov [si], al ; Data1 in SI

mov [si-1], al ; Data2 in one location less

over: loop inner ; Label over for inner loop

dec bx ; Decrement outer counter

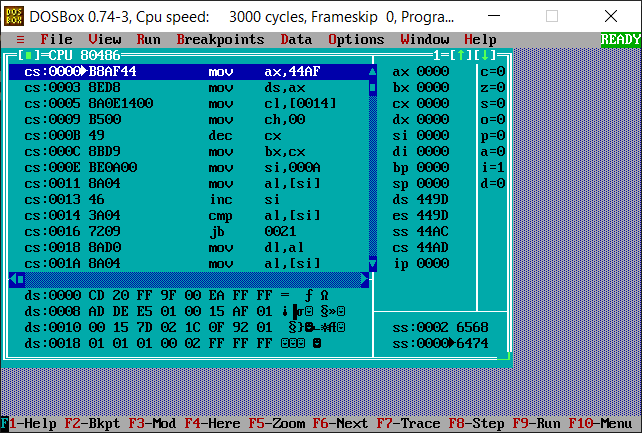
jnz outer ; If counter not zero then repeat outer loop

mov ah, 4ch ; Terminate program

int 21h

end

**Output**:



**Conclusion**: Thus, we have studied and understood the program to sort the given integer array.