**Assignment 1**

**Exercise 1: Find the max element in an array**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **23** | **65** | **12** | **87** | **12** | **343** | **098** | **12** | **76** | **312** |

* **Pre-Condition:**
  + - A variable **max** store the First index in array
    - I plan to compare the whole array index with first index to find the maximum value
* **Post-Condition:**
  + - After the Loop termination, the values of **a** must be comparing with the value in variable **max** and print the greatest one in the array.
* **Loop variant:**
  + - The loop should be terminating once the values in variable **a** is greater than the variable **max.**
* **Loop invariant:**
  + - In the iteration, I initialize the variable **i=0** and with given condition **(i<11).**
    - Using if condition the values in the **a** compared with the value in **max.**Once it reached end of array the maximum value should be printed.
    - In for loop the variable **i** should be incremented.
    - After the n’th iteration, the variable **max** provide the greatest value in an array.

Pseudocode:

Class Maximum

{

int a[10] = {23,65,12,87,12,343,98,12,76,312};

int max, i;

int max = a[0];

for ( i = 0; i < 11; i++)

{

if(a[i]>max);

max = a[i];

}

Print (“ the max no “)}

**Exercise#2: Move zeroes to the end of Array using two pointers technique**

* **Pre-Condition:**

* + - An array **a[6]** store the some values with zeros in an array
    - I am planning to move all zeros to right with the help of two pointers technique
    - Initializing the right and left pointer to zero
    - Creating a **temp** variable to shift the values to left side
* **Post-Condition:**
  + - After the Loop termination, the zeros must be shifted to the right(end) of the array and all the non-zero and non-negative numbers will be in the left side of the array in the same order prior to shifting process
* **Loop variant:**
  + - According to two pointers technique the main loop must terminate only *if the right pointer in the index goes out of array*
* **Loop invariant:**
  + - In this iteration, while loop is placed and it will allow only if the right (index) is less than array length (6)
    - Then if condition will check whether the right index is having zeroes or non zeros
    - *If right pointer sees zero then it will skip the value, if it sees non-zero value then both right and left pointer will make an increment .*
    - The value in the right pointer will be shifted to the left pointer
    - After the n’th iteration, all the non-zero value will be shifted to left of an array and all zeros will be segregated in right

**Pseudo code**

Int a[6]= {0,2,5,0,8,5} ,left =0 , right=0;

Int temp ;

While ( right<6)

{

If ( a(right)=0)

++right;

Else

{

temp=a(left);

a(left)= a(right);

a(right)=temp;

++left;

++right;

}

}