**Question 1 .**

**Write the function for insertion sort.**

**Solution ;**

/\* Function to sort an array using insertion sort\*/

void insertionSort(int arr[], int n)

{

int i, key, j;

for (i = 1; i < n; i++) {

key = arr[i];

j = i - 1;

/\* Move elements of arr[0..i-1], that are

greater than key, to one position ahead

of their current position \*/

while (j >= 0 && arr[j] > key) {

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = key;

}

}

**Question 2.**

**Write a function to find the maximum element in the stack.**

**Solution;**

class StackWithMax

{

// main stack

stack<int> mainStack;

// stack to keep track of max element

stack<int> trackStack;

public:

void push(int x)

{

mainStack.push(x);

if (mainStack.size() == 1)

{

trackStack.push(x);

return;

}

// If current element is greater than

// the top element of track stack, push

// the current element to track stack

// otherwise push the element at top of

// track stack again into it.

if (x > trackStack.top())

trackStack.push(x);

else

trackStack.push(trackStack.top());

}

int getMax()

{

return trackStack.top();

}

int pop()

{

mainStack.pop();

trackStack.pop();

}

};

**Question 3.**

**Write a function to find the minimum element in the stack**

**Solution;**

void push(int x)

    {

        // Insert new number into the stack

        if (s.empty())

        {

            minEle = x;

            s.push(x);

            cout <<  "Number Inserted: " << x << "\n";

            return;

        }