1. Write the function for insertion sort.

**Program:**

void insertionSort(int arr[], int n)

{

    int i, key, j;

    for (i = 1; i < n; i++) {

        key = arr[i];

        j = i - 1;

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

            arr[j + 1] = arr[j];

            j = j - 1;

        }

        arr[j + 1] = key;

    }

}

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

 void push(int x)

    {

        mainStack.push(x);

        if (mainStack.size() == 1)

        {

            trackStack.push(x);

            return;

        }

        if (x > trackStack.top())

            trackStack.push(x);

        else

            trackStack.push(trackStack.top());

    }

    int getMax()

    {

        return trackStack.top();

    }

    int pop()

    {

        mainStack.pop();

        trackStack.pop();

    }

1. Write a function to find the minimum element in the stack

void getMin()

    {

        if (s.empty())

            cout << "Stack is empty\n";

        // variable minEle stores the minimum element

        // in the stack.

        else

            cout <<"Minimum Element in the stack is: "

                 << minEle << "\n";

    }