**QUESTION 1 ANSWER**:

#**include <math.h>**

**#include <stdio.h>**

**/\* 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;**

**}**

**}**

**void printArray(int arr[], int n)**

**{**

**int i;**

**for (i = 0; i < n; i++)**

**printf("%d ", arr[i]);**

**printf("\n");**

**}**

**int main()**

**{**

**int arr[] = { 12, 11, 13, 5, 6 };**

**int n = sizeof(arr) / sizeof(arr[0]);**

**insertionSort(arr, n);**

**printArray(arr, n);**

**return 0;**

**}**

**QUESTION 2 ANSWER :**

**//function to find the maximum element in the //stack.**

**void maxstk()**

**{int i,t;**

**if (top>=0)**

**{ for (int j=0;j<top;j++)**

**{for ( i=0;i<top-j;i++)**

**{**

**if((stk[i]>stk[i+1]))**

**{**

**t=stk[i];**

**stk[i]=stk[i+1];**

**stk[i+1]=t;**

**}**

**}**

**}**

**printf ("\n maximum element in stack is %d",stk[top]);**

**}**

**else**

**{ printf ("\n stack empty \n");}**

**}**

**QUESTION 3 ANSWER :**

**//function to find the minimum element in the //stack**

**void minstk()**

**{ int i,t;**

**if (top>=0)**

**{ for (int j=0;j<top;j++)**

**{for ( i=0;i<top-j;i++)**

**{**

**if((stk[i]<stk[i+1]))**

**{**

**t=stk[i];**

**stk[i]=stk[i+1];**

**stk[i+1]=t;**

**}**

**}**

**}**

**printf ("\n minimum element in stack is %d",stk[top]);**

**}**

**else**

**{ printf ("\n stack empty \n");}**

**}**