Sorting using stack

Code:

//Sort the elements using stack

#include <stdio.h>

#define MAX 100

int pst[MAX],sst[MAX],a[MAX];

int pt=-1,st=-1;

//Considered two stacks - one primary and the other secondary

void push\_p(int d){

    pst[++pt]=d;

}

int pop\_p(){

    int item;

    if(pt!=-1){

        item=pst[pt--];

        return item;

    }

}

void push\_s(int d){

    sst[++st]=d;

}

int pop\_s(){

    int item;

    if(st!=-1){

        item=sst[st--];

        return item;

    }

}

int main(){

    int n,d;

    printf("\nEnter the number of elements to sort:");

    scanf("%d",&n);

    //Taking input in an array

    for(int i=0;i<n;i++){

        printf("\nEnter element:");

        scanf("%d",&a[i]);

    }

    //Pushing the first element of array in the primary stack

    push\_p(a[0]);

    for(int i=1;i<n;i++){

        if(a[i]<=pst[pt]){

            //Incoming element is less than what is already in the top of the stack

            push\_p(a[i]);

        }

        else{

            //Incoming element is greater than what is already in the top of the stack

            while(a[i]>pst[pt]&&pt!=-1){

                //Pop element from primary st and push into secondary st

                d=pop\_p();

                push\_s(d);

            }

            //Finally push the element in primary st

            push\_p(a[i]);

            //Now pop off every elements from primary stack and push into primary stack

            while(st!=-1){

                d=pop\_s();

                push\_p(d);

            }

        }

    }

    //The resultant primary stack is sorted

    printf("\nSorted elements:");

    while(pt!=-1){

        d=pop\_p(pt);

        printf("%d ",d);

    }

}

Output:

Enter the number of elements to sort:5

Enter element:-1

Enter element:5

Enter element:3

Enter element:9

Enter element:10

Sorted elements:-1 3 5 9 10