Question 1

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

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

long int top=-1;

long int stack[100000];

void push(long int x){

top++;

stack[top]=x;

}

void pop(){

top--;

}

void maxElement(){

printf("%ld\n", stack[top]);

}

long int max(long int x, long int y){

return (x>=y ? x:y);

}

int main(void){

long int n, type, x;

scanf("%ld", &n);

for(long int i=0; i<n; i++){

scanf("%ld", &type);

switch(type){

case 1: scanf("%ld", &x);

if(top==-1) push(x);

else push(max(stack[top], x));

break;

case 2:

pop();

break;

case 3:

maxElement();

break;

}

}

return 0;

}

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

struct stack

{

    int data[100000];

    int top;

}st;

int max=0;

void push()

{

    int item;

    scanf("%d",&item);

    st.top++;

    st.data[st.top]=item;

    if(max<st.data[st.top])

        max=st.data[st.top];

}

void pop()

{

    int i;

    if(max == st.data[st.top])

        max=0;

    st.top--;

     for(i=st.top;i>=0;i--)

        if(max<st.data[i])

            max=st.data[i];

}

int main()

{

    int t,n,choice;

    st.top = -1;

    scanf("%d",&t);

    while(t--)

    {

        scanf("%d",&choice);

        switch(choice)

        {

            case 1 :push();

            break;

            case 2:pop();

            break;

            case 3: printf("%d\n",max);

            break;

        }

    }

    return 0;

}

Question 2

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

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

long int top=-1;

long int stack[100000];

void push(long int x){

top++;

stack[top]=x;

}

void pop(){

top--;

}

void minElement(){

printf("%ld\n", stack[top]);

}

long int min(long int x, long int y){

return (x<=y ? x:y);

}

int main(void){

long int n, type, x;

scanf("%ld", &n);

for(long int i=0; i<n; i++){

scanf("%ld", &type);

switch(type){

case 1: scanf("%ld", &x);

if(top==-1) push(x);

else push(max(stack[top], x));

break;

case 2:

pop();

break;

case 3:

minElement();

break;

}

}

return 0;

}