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;

}

**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(min(stack[top], x));

break;

case 2:

pop();

break;

case 3:

minElement();

break;

}

}

return 0;

}