

Е. Максимум в скользящем окне

Ограничение времени	0.6 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

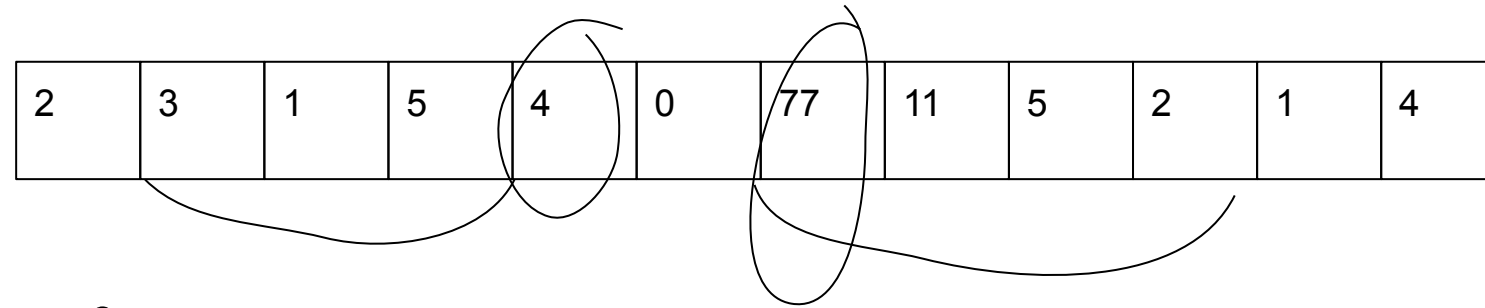
Пусть задан массив из n целых чисел. По этому массиву будут ходить два указателя l и r ($1 \leq l, r \leq n$). Изначально оба они указывают на первый элемент массива ($l = r = 1$). Оба указателя могут двигаться только вправо, на одну позицию за раз. При этом указатель l никогда не оказывается правее указателя r , и ни один из них не выходит за пределы массива. Вам нужно после каждого перемещения указателя определить максимум всех элементов от указателя l вправо до указателя r (включая позиции, на которые указывают l и r).
 Указание. Учетная стоимость обработки каждого запроса на перемещение и подсчет максимума должна оказаться $O(1)$.

Формат ввода

В первой строке входного потока задано число n ($1 \leq n \leq 100\,000$) — размер массива. Во второй строке n целых чисел от $-1\,000\,000\,000$ до $1\,000\,000\,000$ — сам массив. В третьей строке указано число m ($0 \leq m \leq 2n - 2$) — количество перемещений. В четвертой строке — m символов L или R, разделенных пробелами. L означает, что нужно сдвинуть l вправо, R — что нужно сдвинуть r вправо.

Формат вывода

Выведите в одну строку ровно m чисел, где i -е число — максимальное значение на отрезке от l до r после выполнения i -й операции.



$r=0$
 $l=0$

R R R L L L R L R L L R R L (m символов)

- $i=0$ $l=0$ $r=0$ -> $max=2$
- $i=1$ $l=0$ $r=1$ -> $max=3$
- $i=2$ $l=0$ $r=2$ -> $max=3$
- $i=3$ $l=0$ $r=3$ -> $max=5$
- $i=4$ $l=1$ $r=3$ -> $max=5$
- $i=5$ $l=2$ $r=3$ -> $max=5$
- $i=6$ $l=3$ $r=3$ -> $max=5$
- $i=7$ $l=3$ $r=4$ -> $max=5$
- $i=8$ $l=4$ $r=4$ -> $max=4$
- $i=9$ $l=4$ $r=5$ -> $max=4$
- $i=10$ $l=5$ $r=5$ -> $max=0$
- $i=11$ $l=5$ $r=6$ -> $max=77$
- $i=12$ $l=5$ $r=7$ -> $max=77$

- $i=0$ $l=0$ $r=0$ $max=2$
- $i=1$ $l=0$ $r=1$ $max=3$
- $i=2$ $l=0$ $r=2$ $max=3$
- $i=3$ $l=0$ $r=3$ $max=5$
- $i=4$ $l=1$ $r=3$ $max=5$
- $i=5$ $l=2$ $r=3$ $max=5$
- $i=6$ $l=3$ $r=3$ $max=5$
- $i=7$ $l=3$ $r=4$ $max=5$
- $i=8$ $l=4$ $r=4$ $max=4$
- $i=9$ $l=4$ $r=5$ $max=4$
- $i=10$ $l=5$ $r=5$ $max=0$
- $i=11$ $l=5$ $r=6$ $max=77$
- $i=12$ $l=5$ $r=7$ $max=77$
- $i=13$ $l=5$ $r=7$ $max=77$
- $i=14$ $l=6$ $r=7$ $max=77$

```

#include <iostream>
using namespace std;
void printar(int mas[], int length)
{
    for(int i=0;i<length;i++)
    {
        cout<<mas[i]<<" ";
    }
    cout<<endl;
}
void moving_window()
{
    int mas[]={2,3,1,5,4,0,77,11,5,2,1,4};
    int mas_r[]={0,0,0,1,1,1,1,1,1,0,1,0,1,0,1,0,1};
    int r=0, l=0, temp_max, flag=0;
    for(int i=0;i<14;i++)
    {
        flag=0;
        for(int u=1;u<=r+u++;)
        {
            if(flag=0)
            {
                temp_max=mas[u];
                flag=1;
            }
            else if(flag=1 && mas[u]>temp_max)
            {
                temp_max=mas[u];
            }
        }
        cout<<"*c1c1c1c" < i < "c1c1c1c" < u < "c1c1c1c" < mas < "c1c1c1c" < temp_max < endl;
        if(mas_r[i]==0)
        {
            r++;
        }
        else if(mas_r[i]==1 && i<14)
        {
            l++;
        }
    }
    for(int u=1;u<=r+u++;)
    {
        if(flag=0)
        {
            temp_max=mas[u];
            flag=1;
        }
        else if(flag=1 && mas[u]>temp_max)
        {
            temp_max=mas[u];
        }
    }
    cout<<"*c1c1c1c" < i < "c1c1c1c" < u < "c1c1c1c" < mas < "c1c1c1c" < temp_max < endl;
}
void moving_window_v2()
{
    int mas[]={2,3,1,5,4,0,77,11,5,2,1,4};
    int mas_r[]={0,0,0,1,1,1,1,1,1,0,1,0,1,0,1};
    int r=0, l=0, temp_max=mas[0], temp_max2, flag, kolvo1=1, kolvo2=0;
    printar(mas,12);
    for(int i=0;i<14;i++)
    {
        cout<<"*c1c1c1c" < i < "c1c1c1c" < u < "c1c1c1c" < mas < "c1c1c1c" < temp_max < "c1c1c1c" < temp_max2 < "c1c1c1c" < kolvo1 < "c1c1c1c" < kolvo2 < "c1c1c1c" < endl;
        if(mas_r[i]==0)
        {
            r++;
            if(temp_max==mas[r])
            {
                kolvo1++;
            }
            if(kolvo2>0 && temp_max2==mas[r])
            {
                kolvo2++;
            }
            if(temp_max==mas[r])
            {
                temp_max2=temp_max;
                kolvo2=kolvo1;
                temp_max=mas[r];
                kolvo1=1;
            }
            if(kolvo2>0 && temp_max2==mas[r] && mas[r]<temp_max || kolvo2==0 && mas[r]<temp_max)
            {
                temp_max2=mas[r];
                kolvo2=1;
            }
        }
        else if(mas_r[i]==1 && i<14)
        {
            if(mas[l]==temp_max)
            {
                if(kolvo1>1)
                {
                    kolvo1--;
                }
            }
            else
            {
                //cout<<"temp_max2"<temp_max2<<" temp_max"<temp_max2<<endl;
                temp_max2=temp_max;
                kolvo1=kolvo2;
                flag=0;
            }
            for(int u=1;u<=r+u++;)
            {
                if(flag=1 && kolvo2>0 && mas[u]>temp_max2)
                {
                    kolvo2++;
                }
                if(flag=0 && mas[u]>temp_max)
                {
                    temp_max=mas[u];
                    kolvo2=1;
                    flag=1;
                }
            }
            else if(flag=1 && kolvo2>0 && mas[l]>temp_max2 && mas[l]>temp_max)
            {
                temp_max2=mas[l];
                kolvo2=1;
            }
        }
        if(flag=0)
        {
            kolvo2=0;
        }
    }
    cout<<"*c1c1c1c" < i < "c1c1c1c" < u < "c1c1c1c" < mas < "c1c1c1c" < temp_max < endl;
}
int main()
{
    //moving_window();
    moving_window_v2();
    return 0;
}

```