

```

int l = 0, r = 0;
string inp;
for (int i = 0; i < n; ++i)
{
    cin >> inp;
    if (inp == "L") ++l;
    if (inp == "R") ++r;
    if (inp == "P")
    {
        cout << min(l, r) << " ";
        if (l > r)
        {
            cout << "L" << endl;
            l -= r;
        }
        else
        {
            cout << "R" << endl;
            r -= l;
        }
    }
}
cout << l << endl << r << endl;
return 0;
}

```

Задача D2. ЧИСЛО НА ЕДИНГТЪН, автор Красимир Манев

Всеки уважаващ себе си велосипедист трябва да знае собственото си *число на Едингтън*, а то е най-голямото цяло неотрицателно E такова, че поне E пъти в живота си е изминавал с велосипеда си поне E метра без почивка. Естествено, ако един човек никога не е карал велосипед, числото му на Едингтън е нула. Но младият програмист Станчо е и запален велосипедист. Научавайки за числото на Едингтън той решил да пресметне, колко е то в неговия случай. Записал на един лист броя N на случаите, в които е бил на седлото на любимия си „байк“ без почивка и по колко метра е изминал във всеки от тях. Започнал да пресмята, но числата били толкова много, че скоро се забъркал в сметките. Знанията му по програмиране, пък, не били достатъчни и затова оставя на Вас да се справите с проблема. Напишете програма **EDI**, която да пресмята числото на Едингтън.

Вход

На първия ред на стандартния вход ще бъде зададено числото N , $1 \leq N \leq 100000$. Всеки от следващите N реда ще съдържа по едно цяло положително число L , дължината в метри измината при поредното каране $1 \leq L \leq 10000$.

Изход

На стандартния изход програмата трябва да изведе намереното число на Едингтън.

ПРИМЕР

Вход

```

10
3
4
12
5
2
6
2
8
1
7

```

Изход

```

5

```

Решение

Това, че числото N е голямо, е без значение. Решението на задачата, което предлагаме, ще може да работи и с много по-големи стойности на N – милиони и даже милиарди. Важното е, че при всяко каране Станчо е изминавал не повече от 10000 метра. Затова дефинираме масив `int a[10001]` и даваме стойност нула на елементите му (първият цикъл `for`). В него ще натрупаме статистиката за каранията, които е осъществил Станчо – в `a[i]` ще намерим колко пъти е изминавал по i метра (вторият цикъл `for`). Числото на Едингтън намираме в променливата E с цикъла `while`. Първата стойност, която проверяваме е 10000. За да има Станчо число на Едингтън 10000 е необходимо `a[10000] ≥ 10000`. Ако това е така, изпълнението на цикъла ще се прекрати и програмата ще изведе 10000. Ако това не е така, за да има Станчо число на Едингтън 9999 е необходимо `a[9999] + a[10000] ≥ 9999`. Затова, в тялото на цикъла, към съдържанието на `a[E-1]` (в случая `a[9999]`) добавяме съдържанието на `a[E]` (в случая `a[10000]`) и намаляваме стойността на E (след `E--` в променливата E ще имаме 9999). Така ще накараме оператора `while` да сравни броя на каранията от поне 9999 метра с 9999 и т.н. На всяка стъпка на цикъла, при която E още не е търсеното число на Едингтън, ще добавяме общия брой на каранията от поне E метра към броя на тези от $E-1$ метра, за да намерим броя на каранията от поне $E-1$ метра и да го сравним с $E-1$. Тъй като дължините са положителни числа, натрупваната сума ще расте, а числото в E ще намалява и неравенството `a[E] < E`, рано или късно, ще бъде нарушено. Изпълнението на цикъла ще се прекрати и съдържанието на E ще бъде търсеното число на Едингтън.

```

#include <iostream>
using namespace std;
int main()
{
    int a[10001], N, i, L, E;
    cin >> N;
    for (i = 0; i <= 10000; i++) a[i] = 0;

```