

水仙花数

水仙花数是指一个 m 位自然数等于自身各个数位上数字的 m 次幂之和。

例如：153是一个三位的水仙花数， $1^3 + 5^3 + 3^3 = 153$ 。

输入格式

第一行输入一个整数 n ，接下来 n 行，每行输入一个需要判断的整数，其范围属于 $[1, 1e9]$ 。

输出格式

共 n 行，对于每个输入，输出它是否为水仙花数，如果是水仙花数，输出1，否则输出0。。

一般实现（略）

- 注意变量的初始化和重置

通过函数实现

```
#include <iostream>
#include <cmath>
using namespace std;

int num_len(int n) {
    int len = 0;
    while (n > 0) {
        n = n / 10;
        len++;
    }
    return len;
}
```

```
int main() {
    int n, num, temp, sum, len;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> num;
        sum = 0;
        temp = num;
        len = num_len(num);
        while (temp > 0) {
            sum += pow(temp % 10, len);
            temp = temp / 10;
        }
        if (sum == num) cout << 1 << endl;
        else cout << 0 << endl;
    }
    return 0;
}
```

通过函数实现

```
int num_len(int n) {
    int len = 0;
    while (n > 0) {
        n = n / 10;
        len++;
    }
    return len;
}

int num_sum(int n) {
    int sum = 0;
    int len = num_len(n);
    while (n > 0) {
        sum += pow(n % 10, len);
        n = n / 10;
    }
    return sum;
}
```

```
int main() {
    int n, num;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> num;
        if (num_sum(num) == num)
            cout << 1 << endl;
        else
            cout << 0 << endl;
    }
    return 0;
}
```

通过递归函数实现

```
int num_len(int n) {  
    if (n / 10 == 0)  
        return 1;  
    return num_len(n / 10) + 1;  
}
```

```
int num_sum(int n, int l) {  
    if (n / 10 == 0)  
        return pow(n, l);  
    return num_sum(n / 10, l) + pow(n % 10, l);  
}
```

```
int main() {  
    int n, num;  
    cin >> n;  
    for (int i = 0; i < n; i++) {  
        cin >> num;  
        if (num_sum(num, num_len(num)) == num) cout << 1 << endl;  
        else cout << 0 << endl;  
    }  
    return 0;  
}
```

弹奏黑白块

小蓝鲸是一个钢琴爱好者，他喜欢黑键和白键轮流弹奏。

现在他遇到了一串连续的由黑块和白块构成的序列，他想知道他按照序列顺序分别轮流弹奏黑块和白块，最多能弹奏多少下？

在题目中，黑块和白块分别由 **B** 和 **W** 字符来表示。我们需要获得最长的连续交错的 **BWBW...** 或 **WBWB...** 序列。

输入格式

第一行输入一个整数 n ，表示输入序列的长度

第二行输入长度为 n 的由 **B** 和 **W** 字符构成的文本序列。

输出格式

一个整数，表示按照当前弹奏规则最多能弹奏的次数。

```
int n, rest = 0;
char ch, prev_ch;
cin >> n;
getchar();
prev_ch = getchar();
for (int i = 0; i < n; i++) {
    ch = getchar();
    if (ch != prev_ch) {
        rest++;
        prev_ch = ch;
    }
}
cout << rest << endl;
```