

# 积水面积

---

## 人员

周苇杰、康佳、洪晨栋、洪晨棋 到课，罗启宸、李沛都、陶汇笙、邹忆航 线上

## 作业

<https://www.luogu.com.cn/contest/186516>

题单中 A、B、C、D 4道题目为课上内容，要求同学们课下补完

E 题为必做作业，F 题为选做作业

## 课堂表现

同学们课上听讲都很认真，希望课下也要做好复习，把课上题目再多复习几遍。

## 课堂内容

### P1307 [NOIP2011 普及组] 数字反转

```
// while 循环拆数核心代码
int ans = 0;
while (n != 0) {
    int t = n%10;
    n /= 10;
    ans = ans*10 + t;
}
// 循环结束后，ans 变量存储的就是反转后的值
```

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int n, flag = 1;
    cin >> n;
    if (n < 0) {
        n = -n;
        flag = -1;
    }

    // 正: flag==1  负: flag== -1
    int ans = 0;
    while (n != 0) {
```

```
    int t = n%10;
    n /= 10;
    ans = ans*10 + t;
}
cout << flag * ans << endl;
return 0;
}
```

### P1179 [NOIP2010 普及组] 数字统计

变量  $i$  从  $l \sim r$  循环，每次利用 while 循环拆数统计变量  $i$  中有几个 2 就可以了

**注: 不能直接对变量  $i$  进行拆数，因为对变量  $i$  拆数完后变量  $i$  就变为 0 了**

```
#include <bits/stdc++.h>

using namespace std;

int main()
{
    int l, r;
    cin >> l >> r;
    int sum = 0;
    for (int i = l; i <= r; i++) {
        int cnt = 0, x = i;
        while (x != 0) {
            int t = x%10;
            x /= 10;
            if (t == 2) {
                cnt++;
            }
        }
        sum += cnt;
    }
    cout << sum << endl;
    return 0;
}
```

### P1318 积水面积

对于位置  $i$  来说，如何统计位置  $i$  可以积多少水？

取决于 位置  $i$  左边的最大值 和 位置  $i$  右边的最大值 中的最小值

因为对于  $i$  左边来说，如果左边的最大值是  $left\_max$ ，只要位置  $i$  这里最终水的的高度不超过  $left\_max$ ，那么水就不会从左边往外流

同理，如果右边的最大值是  $right\_max$ ，只要位置  $i$  这里最终水的的高度不超过  $right\_max$ ，那么水就不会从右边往外流

最终，我们想让水从两边都不要往外流，所以水的高度既不能超过 left\_max，又不能超过 right\_max

```
#include <bits/stdc++.h>

using namespace std;

int w[10005];

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> w[i];
    }

    int sum = 0;
    for (int i = 1; i <= n; i++) {
        int left_max=-1, right_max=-1;
        for (int j = 1; j <= i-1; j++) {
            if (w[j] > left_max) {
                left_max = w[j];
            }
        }
        for (int j = i+1; j <= n; j++) {
            if (w[j] > right_max) {
                right_max = w[j];
            }
        }
        if (min(left_max, right_max) - w[i] > 0) {
            sum += min(left_max, right_max) - w[i];
        }
    }
    cout << sum << endl;
    return 0;
}
```

#### T485134 按按钮

两个人都会先按 C 类的按钮

然后，第一个人按 A，第二个人按 B

c 是偶数:

- a > b: a赢
- a == b: b赢

- $a < b$ : b 赢

c 是奇数:

- $a++$
- $a > b$ : a 赢
- $a == b$ : a 赢
- $a < b$ : b 赢