

# 高精加 + 高精乘

## 人员

温郝冬、陶汇笙、邹忆航、魏子喆、洪晨栋、洪晨棋、罗启宸、郭栩睿、马瑞昕、郭恩名、李沛都、张曦月、周晨睿 到课

## 作业检查

作业链接: <https://www.luogu.com.cn/contest/207529> (C D E)

郭栩睿、魏子喆 做了 3 道题

洪晨栋、陶汇笙、崔宸赫、郭恩名 做了 2 道题

温郝冬、罗启宸、邹忆航 做了 1 道题

其他同学未做

## 作业

<https://www.luogu.com.cn/contest/211194> (A B C D 4 个题)

## 课堂表现

同学们整体上课听讲比较认真，这节课题目实现较为复杂，课下要好好复习。

## 课堂内容

### U478308 求合法对数

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

using namespace std;

const int maxn = 1e5 + 5;
int a[maxn], b[maxn], c[maxn];
int cnt[maxn]; // cnt[i]: b[c[1]], b[c[2]], ..., b[c[n]] 中 i 出现过多少次

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> a[i];
    for (int i = 1; i <= n; ++i) cin >> b[i];
    for (int i = 1; i <= n; ++i) cin >> c[i];

    for (int i = 1; i <= n; i++) {
        cnt[b[c[i]]]++;
    }
}
```

```
long long res = 0;
for (int i = 1; i <= n; i++) {
    res += cnt[a[i]];
}
cout << res << endl;
return 0;
}
```

## U480698 Wandering

```
// 48 分代码, 此题拿 48 分即可
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
int a[maxn];

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

    LL dis = 0;
    long long maxx = 0;
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= i; j++) {
            dis += a[j];
            maxx = max(maxx, dis);
        }
    }
    cout << maxx << endl;
    return 0;
}
```

## P1601 A+B Problem (高精)

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

using namespace std;

string add(string a, string b) {
    reverse(a.begin(), a.end()); reverse(b.begin(), b.end());
    while (a.size() < b.size()) a += "0";
    while (b.size() < a.size()) b += "0";
    reverse(a.begin(), a.end()); reverse(b.begin(), b.end());
```

```

string res;
int n = a.size(), t = 0;
for (int i = n-1; i >= 0; i--) {
    int x = a[i] - '0';
    int y = b[i] - '0';
    int k = x + y + t;

    t = k/10;
    k %= 10;
    res += char(k+'0');
}

if (t == 1) res += "1";
reverse(res.begin(), res.end());
return res;
}

int main()
{
    string a, b;
    cin >> a >> b;
    cout << add(a, b) << endl;
    return 0;
}

```

```

// 方法二
string add(string a, string b) {
    reverse(a.begin(), a.end()); reverse(b.begin(), b.end());

    string res;
    int t = 0;
    for (int i = 0; i < (int)a.size() || i < (int)b.size() || t; i++) {
        int x = 0, y = 0;
        if (i < (int)a.size()) x = a[i] - '0';
        if (i < (int)b.size()) y = b[i] - '0';

        int k = x + y + t;
        t = k / 10;
        res += (k%10 + '0');
    }
    reverse(res.begin(), res.end());
    return res;
}

```

### P1303 A\*B Problem

```

#include <bits/stdc++.h>

using namespace std;

```

```
const int maxn = 4000 + 5;

string mul(string a, string b) {
    if (a=="0" || b=="0") return "0";

    int c[maxn] = {0}; // c[0]: 个位, c[1]: 十位, c[2]: 百位 .....

    // a: 372   b: 6459
    // 反转
    // a: 273   a[0]: a的个位, a[1]: a的十位, a[2]: a的百位
    // b: 9546   b[0]: b的个位, b[1]: b的十位, b[2]: b的百位
    reverse(a.begin(), a.end());
    reverse(b.begin(), b.end());

    // a[i] * b[j] -> c[i+j]
    for (int i = 0; i < (int)a.size(); i++) {
        for (int j = 0; j < (int)b.size(); j++) {
            c[i+j] += (a[i]-'0') * (b[j]-'0');
        }
    }

    // 把 c 数组进行进位操作: 从 个位 往前进位 -> 从 c[0] 开始
    int t = 0;
    for (int i = 0; i < maxn; i++) {
        int k = c[i] + t;
        t = k / 10;
        c[i] = k % 10;
    }

    // 从 c 数组的最高位开始看, 找到第一个 非0 的数, 这个数就是最高位
    string res;
    bool flag = false; // 找到第一个不是 0 的数, 把 flag 变为 true
    for (int i = maxn-1; i >= 0; i--) {
        if (c[i]!=0 && flag==false) flag = true;
        if (flag == true) {
            res += char(c[i]+'0');
        }
    }

    return res;
}

int main()
{
    string a, b;
    cin >> a >> b;
    cout << mul(a, b) << endl;
    return 0;
}
```