

根号n求因数数量

人员

温郝冬、董浩桢、周苇杰、陶汇笙、张昱霖、郭栩睿、马瑞昕、崔宸赫 到课，李沛都、罗启宸、康佳 线上

作业检查

温郝冬 上周请假

李沛都 未完成

罗启宸 已完成

董浩桢 上周请假

周苇杰 已完成

陶汇笙 已完成

张昱霖 已完成

康佳 未完成

郭栩睿 上周请假

马瑞昕 上周请假

崔宸赫 上周请假

邹忆航 已完成

作业

<https://www.luogu.com.cn/contest/188483>

A、B、C 题必做，D 题选做

课堂表现

董浩桢、陶汇笙 2位同学课上做题比较快比较好，在老师讲完思路后，讲代码之前就可以独立把题目写出来，提出表扬！！

课堂内容

T464056 variable

```
1. cin >> n  
  
2. a*b + c*d == n
```

```

x + y == n
x: 1 ~ n-1 循环
    y = n-x
    求 x 有几个因数 -> p
    求 y 有几个因数 -> q
    对应的这一组 x 和 y,
        总共有 p*q 种合法的 a/b/c/d 的情况
    所以对 p*q 进行累加

```

3. 优化? -> 尝试优化计算因数数量的时间复杂度

```

for (i: 1 ~ 根号n) {
    if (n % i == 0) {
        if (i == 根号n) {
            +1
        }
        else {
            +2
        }
    }
}

```

4. 根号n -> #include <cmath>

```

int t = sqrt(n)
for (int i = 1; i <= t; i++) {

}

```

这种写法不推荐

5.

```

for (int i = 1; i*i <= n; i++) {
    if (n%i == 0) {
        if (i*i == n) {
            +1
        }
        else {
            +2
        }
    }
}

```

```

#include <bits/stdc++.h>

```

```

using namespace std;

```

```

int calc(int x) { // 计算 x 有几个因数
    int res = 0;
    for (int i = 1; i*i <= x; i++) {
        if (x % i == 0) {
            if (i*i == x) {
                res++;
            }
            else {

```

```
        res += 2;
    }
}
return res;
}

int main()
{
    int n;
    cin >> n;
    long long ans = 0;
    for (int x = 1; x <= n-1; x++) {
        int y = n-x;

        int p = calc(x), q = calc(y);
        ans += (long long)p*q;
    }
    cout << ans << endl;
    return 0;
}
```

P1320 压缩技术 (续集版)

两种输入方法：

```
1.
string s, t;
int n;
while (cin >> s) {
    t += s;
    n = s.size();
}

2.
string s, t;
cin >> s;
t += s;
int n = s.size();
for (int i = 1; i <= n-1; i++) {
    cin >> s;
    t += s;
}
```

之后，就是统计字符串 `t` 中有几个连续的 `0`，几个连续的 `1` 了，跟之前做过的题目比较相似
维护一个 `char` 字符 `c` 代表前一个字符是什么，维护一个整数 `cnt` 代表字符 `c` 连续出现了多少次
然后 `i` 从 `1 ~ t.size()-1` 循环，每次判断 `t[i]` 跟字符 `c` 相不相同即可

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

using namespace std;

int main()
{
    string s, t;
    int n;
    while (cin >> s) {
        t += s;
        n = s.size();
    }
    cout << n << " ";

    int len = t.size();
    char c = '0';
    int cnt = 0; // 字符 c 出现了 cnt 次
    for (int i = 0; i < len; i++) {
        if (s[i] == c) {
            cnt++;
        }
        else {
            cout << cnt << " ";
            c = s[i];
            cnt = 1;
        }
    }
    cout << cnt << endl;
    return 0;
}
```