

# 1. 同构数

最右端几位数？

所谓同构数，指的是一个特殊的正整数，恰好出现在其平方数的**最右端**。请你设计C/C++程序，输入一个正整数n，输出正整数区间 [1, n] 所有同构数的个数。

$$1*1=1$$

$$25*25 = 625$$

$$376*376 = 141376$$

求余数

除数是变量

```
int Isomorph(int i)
{
    if(i < 10 && i == i*i % 10)
        return 1;
    else if(i < 100 && i == i*i % 100)
        return 1;
    else if(i < 1000 && i == i*i % 1000)
        return 1;
    .....
    return 0;
}
```

# 第一种思路

```
int n, i = 1;
cin >> n;

while(n >= 10)
{
    ++i;
    n /= 10;
}
...
```

n 是几位数? i    循环、除法

```
bool IsomorphM(int n, int i)
{
    if(n == n*n % pow(10, i))
        return true;
    else
        return false;
}
```

# 第一种思路

```
int n, i = 0;  
cin >> n;
```

```
while(n > 0)  
{  
    ++i;  
    n /= 10;  
}  
...
```

n 是几位数? i    循环、除法

n >= 0 会怎样?

```
bool IsomorphM(int n, int i)  
{  
    if(n == n*n % pow(10, i))  
        return true;  
    else  
        return false;  
}
```

# 第一种思路（拓展）

```
int n, i = 1;  
cin >> n;
```

```
while(n >= 10)  
{  
    ++i;  
    n /= 10;  
}  
...
```

n 是几位数？ i    循环、除法

$10^i$

循环、乘法

```
bool IsomorphM(int n, int i)  
{  
    if(n == n*n % pow(10, i))  
        return true;  
    else  
        return false;  
}
```

# 第一种思路（拓展）

```
int n, i = 1;  
cin >> n;
```

```
while(n > 0)  
{  
    i *= 10;  
    n /= 10;  
}  
...
```

n 是几位数？ i

循环、除法

$10^i$

循环、乘法

两个循环可以合并

# 第一种思路（拓展）

```
int n, i = 1;
cin >> n;
int temp = n;
while(temp > 0)
{
    i *= 10;
    temp /= 10;
}
if(IsomorphM(n, i))
    ...
```

n 是几位数？ i    循环、除法

循环、乘法

两个循环可以合并

```
bool IsomorphM(int n, int i)
{
    if(n == n*n % i)
        return true;
    else
        return false;
}
```

## 第二种思路

n 是几位数？ 不重要

一起循环、除法、逐位判断

$$1*1=\textcolor{red}{1}$$

$$25*25 = 6\textcolor{red}{25}$$

$$376*376 = 141\textcolor{red}{376}$$

```
bool IsomorphM(int n)
{
    int m = n * n;
    while (n > 0) {
        if (n % 10 != m % 10)
            return false;
        n /= 10;
        m /= 10;
    }
    return true;
}
```

## 2. 四方定理

---

数论中的“四方定理”指的是一个正整数**至多**只要用四个整数的平方和就可以表示。请你设计C/C++程序，统计一个正整数可以这样表示的方案数目。要求在main函数中输入一个正整数n，输出其对应的方案数目，用另一个函数实现方案数目的统计。

**注意：**像 $9 = 2 \times 2 + 2 \times 2 + 1 \times 1$ 和 $9 = 1 \times 1 + 2 \times 2 + 2 \times 2$ 这样等仅顺序不同的方案视为同一种方案。



## 有序、剪枝

## 有序、剪枝

## 有序、剪枝

## 有序、剪枝

### 3. 礼炮声声

---

在某典礼上，甲、乙、丙三门礼炮要同时开始鸣放，并分别鸣放  $n$  响，其中甲礼炮每3秒放一次，乙礼炮每4秒放一次，丙礼炮每5秒放一次。假定没有哑炮，且多门礼炮同时鸣放时听众只能听到一次礼炮声，那么当三门礼炮都鸣放完毕时，听众总共可以听到几次礼炮声？请你设计C/C++程序实现计算任务。

去掉重复声音？还是枚举听到的声音？

# 思路一：正向枚举听到的声音

---

```
int sum = 0, i = 0;
while (i <= 3 * n - 3) {
    if (i % 3 == 0 || i % 4 == 0 || i % 5 == 0)
        sum += 1;
    i += 1;
}
while (i <= 4 * n - 4) {
    if (i % 4 == 0 || i % 5 == 0)
        sum += 1;
    i += 1;
}
while (i <= 5 * n - 5) {
    if (i % 5 == 0)
        sum += 1;
    i += 1;
}
```

## 思路二：减去重叠声音

---

```
int n, m = 0;
cin >> n;
m = 3 * n;
for (int i = 0; i <= 3 * (n - 1); i += 3) {
    if (i % 4 == 0 || i % 5 == 0)
        m--;
}
for (int i = 0; i <= 4 * (n - 1); i += 4) {
    if (i % 5 == 0)
        m--;
}
cout << m;
```

甲礼炮每3秒放一次，有乙丙存在重叠

乙礼炮每4秒放一次，有丙存在重叠

## 思路二：减去重叠声音（第一个版本的思路）

```
for (int i = 0; i <= 3 * (n - 1); i += 3) {  
    if (i % 4 == 0)  
        m--;  
    if (i % 5 == 0)  
        m--;  
}
```

```
for (int i = 0; i <= 4 * (n - 1); i += 4) {  
    //减多了，多减了 $[0, 3 * (n - 1)]$ 范围内的 $3*4*5$ 三声重叠的次数  
    if (i % 5 == 0)  
        m--;  
}
```

$m += (3 * (n - 1) / (3 * 4 * 5)) + 1;$  // +1是因为0的时候至少也有一声重叠

## 4. 数据恢复

---

小蓝鲸将一批重要的数据存在电脑里，但是很不幸，这天小蓝鲸的电脑系统受到了奇怪的干扰，这批数据的高位和低位发生了交换。已知数据中的最大值不超过无符号整型数据范围，并且每个数交换的两部分是高16位和低16位。你能用C/C++代码实现数据的恢复吗？

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

int main() {
    unsigned int n;
    cin >> n;
    cout << (n << 16) + (n >> 16) << endl;
    return 0;
}
```

## 5. [选做] 次方相乘

---

请设计C/C++程序，实现将一个大于1的正整数  $n$  表示成所有素数因子的次方相乘的形式输出，次方用英文圆括号()表示。要求按从小到大的顺序输出素数因子。



# 思路一

```
int n;
scanf("%d", &n);
int count;
int a = n;
for (int i = 2; i <= sqrt(n); i++) {
    int isprime = 1;
    for (int j = 2; j * j <= i; j++) {
        if (i % j == 0)
            isprime = 0;
        if (isprime == 0)
            break;
    }
```



容易超时!

```
        if (isprime == 1) {
            count = 0;
            while (a % i == 0) {
                a = a / i;
                count++;
            }
            if (count != 0)
                printf("%d(%d)", i, count);
        }
    }
    if (a > 1) printf("%d(%d)", a, 1);
```

# 思路二

---

```
int n;  
scanf("%d", &n);  
for (int i = 2; i <= n; i++)  
{  
    int j = 0;  
    while (n % i == 0)  
    {  
        j += 1;  
        n = n / i;  
    }  
    if (j != 0)  
        printf("%d(%d) ", i, j);  
}
```