

2019程序设计基础 第一次上机测验 测试讲解

郭延文 李元琪 费炆 李恣炜

2019-11-11

评分标准

- 每题5分，共计15分，量化为百分制后的10分
- $\max(\text{测试用例分数}n(5), \text{按思路给分})$
- 按思路给分：
 - 编译不过扣一分/严重超时扣一分
 - 思路有小问题扣两分
 - 思路完全不对扣三分

1. 计算闰年

计算闰年 难度：简单

输入两个年份，均为四位数字，输出两年之间（包含这两年）闰年的数量。

示例输入：

1999 2001

输出：

1

示例输入：

1898 2006

输出：

26

示例输入：

2001 2006

输出：

1

示例输入：

1999 2008

输出：

3

计算闰年 参考代码

```
#include<iostream>
using namespace std;
int main()
{
    int a, b;
    cin >> a >> b;
    int count = 0;
    for (int i = a; i <= b; i++)
    {
        if (i % 4 == 0)
        {
            if (i % 100 != 0 || i % 400 == 0)
                count++;
        }
    }
    cout << count;
    return 0;
}
```

另外一种思路：

1. 年份相减除以4
2. 再判断100和400的特殊

情况，做减法

典型问题

- `cin >> a, b;`

这种写法根本就是错的，要输入两个值`cin >> a>>b;`

- `while (x == 0, y != 0, z == 0)`

连接条件，应该用`&&`，`||`，`!`表示与，或，非

```
while (x == 0&&y != 0&&z == 0)
{
    n = n + 1;
}
```

死循环！`while`的循环里如果没有修改循环条件参数的操作，一旦进去就不可能终止。要判断，用`if`

2. 求最大质因数

求一个数的最大质因数。

例如13195的所有质因数为5、7、13、29，则最大质因数为29。

提示：使用long long类型。

输入：

一个实数x

输出：

x的最大质因数

样例输入：

13195

样例输出：

29

测试用例：

input:

13195

output:

29

input:

167076

output:

17

input:

600851475143

output:

6857

input:

10789

output:

10789

最大质因数 参考代码

- 质因数分解，不能采取简单遍历，否则大的数据求不出来
- 输入 m ，每次找到一个质因数 x 以后，接下来就应该用 m/x 来找下一个质因数

```
#include<iostream>
using namespace std;
int main(){
    long long num,x = 2,ans = 0;
    cin >> num;
    while(num != 1){
        while(num % x != 0){
            x++;
        }
        ans = x;
        // 除掉 num 中的所有值为 x 的素因子
        while(num % x == 0){
            num /= x;
        }
        x++;
    }
    cout << ans << endl;
}
```

典型问题

```
for (long long int i = 3; i <= x / 3; i += 2)
{
    if (x % i == 0)
    {
        long long int m = 0;
        for (long long int j = 1; j < sqrt(i); j += 2)
        {
            if (i % j == 0)
                m++;
        }
        if (m == 1)
            y = i;
    }
}
```

- 这样遍历，对 longlongint 肯定是不行的，不仅oj上过不了，vs上我也跑不出结果，六位数以内还行，七位数就跑不出了
- 当然，这个同学的遍历方法，没办法统计到最大质因数是它本身的情况

- 也不要忽略了这个数本身就是质数的情况，这时候，输出就应该就是这个数本身
- 输入10789 输出10789

```
int main() {  
    long long int n;  
    int max = 1;  
    cin >> n;  
    for (int i = 1; i <= n / 2; i += 2) {  
        int j = 2;  
        while (n % i == 0 && i % j != 0 && j < i) {  
            j++;  
        }  
        if (j == i)  
            max = (max >= i) ? max : i;  
    }  
    cout << max << endl;  
    system("pause");  
    return 0;  
}
```

```
12 int main()
13 {
14     do {
15         long long int n;
16         cin >> n;
17         int ans;
18         for (long long int i = 1; i <= n; i++) {
19             if (n % i == 0) {
20                 if (judge(i)) {
21                     cout << i << endl;
22                     break;
23                 }
24             }
25         }
26     } while (true);
27 }
```

最大质因数：这里面漏
掉了判断最大

3. 例：计算级数和的值

- 编写程序计算当 $x=0.5$ 时下述级数和的近似值，使其误差（前后两次级数和的差）小于某一指定的值ZERO(例如：ZERO=0.000001)

$$x - (x^3)/(3*1!) + (x^5)/(5*2!) - (x^7)/(7*3!) + \dots$$

即：

$$x - \frac{x^3}{3*1!} + \frac{x^5}{5*2!} - \frac{x^7}{7*3!} + \dots$$

PS: 自己可以试验不同的精度，看循环次数和收敛情况

参考代码

这道题debug

的时候，分别

确定阶乘计算

的是否正确。

然后手动计算

几项

```
#include<iostream>
#include<cmath>
using namespace std;
const double ZERO = 0.000001;
int factorial(int n)
{
    if (n == 1) return 1;
    return n * factorial(n - 1);
}
int main()
{
    double x; cin >> x; double s = x;
    for (int i = 1;; ++i) {
        double ans = pow(-1, i) * pow(x, 2 * i + 1) / ((2 * i + 1)
*        factorial(i));
        s += ans;
        if (fabs(ans) < ZERO) break;
    }
    cout << s << endl;
    system("pause");
    return 0;
}
```

典型问题

写反循环条件。在差小于阈值的时候，应该跳出循环

```
while (abs(x2 - x1) < 0.000001)
```

```
{
```

```
    x2 = x1;
```

```
    x1 = x2 + pow(-1, (i - 1) / 2) * pow(x2, i) / (i * jc(i));
```

```
    i += 2;
```

```
}
```

```
double jc(double x)
{
    if (x == 1)
        return 1;
    if (x == 2)
        return 2;
    else
        return jc(x - 1) * jc(x - 2);
}
```

这样写阶乘，很容易在数字大的时候调用堆栈溢出。递归函数不适合太深。这里显然更适合用循环。

判断double类型的相等，最好不要用==，用差的绝对值小于一个很小的数，来判断

Q & A

