

C++方向编程题答案

第四周

day20

题目ID : 36836-字符串反转

链接 : <https://www.nowcoder.com/practice/e45e078701ab4e4cb49393ae30f1bb04?tpId=37&&tqId=21235&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking>

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

string reverseString(string s) {
    if (s.empty())
        return s;
    size_t start = 0;
    size_t end = s.size() - 1;
    while (start < end)
    {
        swap(s[start], s[end]);
        ++start;
        --end;
    }
    return s;
}

int main()
{
    string s;
    getline(cin,s);
    cout<<reverseString(s)<<endl;
    return 0;
}
```

36884-查找组成一个偶数最接近的两个素数

链接 : <https://www.nowcoder.com/practice/f8538f9ae3f1484fb137789dec6eedb9?tpId=37&&tqId=21283&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking>

```
/*
    素数：除过1和本身，不能被其它数整除
    最近的两个素数应该从最中间的位置开始向两边查找
*/

#include<iostream>
#include<algorithm>

using namespace std;
```

```
//判断素数
bool isPrime(int num)
{
    int tmp = sqrt(num);
    //只遍历到开方出，不能被开方左边的数整除，则一定不能被开方右边的数整除
    for (int i = 2; i <= tmp; i++)
    {
        if (num % i == 0)
            return false;
    }
    return true;
}

int main()
{
    int num;
    int half;
    int i;
    while (cin >> num)
    {
        half = num / 2;
        //从中间向两边找
        for (i = half; i > 0; i--)
        {
            if (isPrime(i) && isPrime(num - i))
                break;
        }

        cout << i << endl << num - i << endl;
    }
    return 0;
}
```