C++方向编程题答案

第二周

day12

> **题目ID：24951-二进制插入**
>
> 链接：https://www.nowcoder.com/practice/30c1674ad5694b3f8f0bc2de6f005490?tpId=8&&tqId=11019&rp=1&ru=/activity/oj&qru=/ta/cracking-the-coding-interview/question-ranking
>
> **【题目解析】：**
>
> 本题考察位运算
>
> **【解题思路】：**
>
> i j
> m:1024：100000000 00
> n:19 ： 10011
> 要把n的二进制值插入m的第j位到第i位，只需要把n先左移j位，然后再进行或运算（|）即可。
>
> m: 10000000000
>
> n: 00001001100
>
> 10001001100

```
class BinInsert {
public:
int binInsert(int n, int m, int j, int i) {
  m <<= j;
  return n | m;
}
};
```

> **题目ID：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
>
> **【题目解析】：**
>
> 本题题意明确，不做解析
>
> **【解题思路】：**
>
> 本题首先需要判断素数，素数表示除过1和本身，不能被其它数整除。通过循环遍历来判断一个数是否为素数。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;
}
```