

C++方向编程题答案

第二周

day7

题目ID: 45846 --Fibonacci数列

链接: <https://www.nowcoder.com/practice/18ecd0ecf5ef4fe9ba3f17f8d00d2d66?tpId=85&&tqId=29846&rp=1&ru=/activity/oj&qru=/ta/2017test/question-ranking>

```
/*
    思路：先找到距离N最近的两个Fibonacci数
    然后在这两个数中找到距离N最近的数
*/

#include <iostream>
using namespace std;
int main(){
    int N,l,r,f0=0,f1=1,f;
    cin >> N;
    while(1){
        f = f0 + f1;
        f0 = f1;
        f1 = f;
        //找到比N小且距离N最近的数，求出距离
        if(f < N) l = N-f;
        else{
            //找到比N大且距离N最近的数，求出距离
            r = f - N;
            break;
        }
    }
    //取最小距离
    cout << min(l,r) << endl;
    return 0;
}
```

36939-合法括号序列判断

<https://www.nowcoder.com/practice/d8acfa0619814b2d98f12c071aef20d4?tpId=8&&tqId=11039&rp=1&ru=/activity/oj&qru=/ta/cracking-the-coding-interview/question-ranking>

```
/*
    用栈结构实现，栈中存放左括号，当遇到右括号之后，检查栈中是否有左括号
    如果有则出栈，如果没有，则说明不匹配。
*/

class Parenthesis {
public:
```

```
bool chkParenthesis(string A, int n) {  
    // write code here  
    stack<char> sc;  
    for (auto ele : A) {  
        switch (ele) {  
            case '(':  
                sc.push(ele);  
                break;  
            case ')':  
                {  
                    if (sc.empty() || sc.top() != '(')  
                        return false;  
                    else  
                        sc.pop();  
                }  
                break;  
            default:  
                return false;  
        }  
    }  
    return true;  
};
```