C++方向编程题答案

第三周

day17

> 题目ID：36877-杨辉三角的变形
>
> 链接：https://www.nowcoder.com/practice/8ef655edf42d4e08b44be4d777edbf43?tpId=37&&tqId=21276&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking

```
/*
    第n行有2n - 1个元素
    第i,j元素等于上一行第j - 2,j - 1,j三列元素之和
    每一行的第一列和最后一列都为1

*/
#include<iostream>
#include<string>
#include<vector>

using namespace std;
int main()
{
    int n, m;
    while (cin >> n)
    {
        m = 2 * n - 1;
        vector<vector<int>> dp(n, vector<int>(m, 0));
        dp[0][0] = 1;
        for (int i = 1; i < n; i++)
        {
            //第一列和最后一列都为1
            dp[i][0] = dp[i][2 * i] = 1;
            for (int j = 1; j < 2 * i; ++j)
            {
                if (j == 1)
                    //如果是第二列，则只是两个元素之和
                    dp[i][j] = dp[i - 1][j - 1] + dp[i - 1][j];
                else
                    //第i,j元素等于上一行第j - 2,j - 1,j三列元素之和
                    dp[i][j] = dp[i - 1][j - 2] + dp[i - 1][j - 1] + dp[i - 1][j];
            }
        }
        int k;
        for (k = 0; k < m; k++)
        {
            if (dp[n - 1][k] % 2 == 0 && dp[n - 1][k] != 0)
            {
                cout << k + 1 << endl;

                break;
```

```
            }
        }
        if (k == m)
            cout << -1 << endl;
    }
    return 0;
}
```

36902-超长正整数相加

链接：https://www.nowcoder.com/practice/5821836e0ec140c1aa29510fd05f45fc?tpId=37&&tqId=21301&rp=1&ru=/activity/oj&qru=/ta/huawei/question-ranking

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

string addStrings(string num1, string num2) {
        //由低位向高位相加
        int i = num1.size() - 1;
        int j = num2.size() - 1;
        string result = "";
        //当前位的相加结果
        int carry = 0;
        while (i >= 0 || j >= 0) {
            if (i >= 0) {
                carry += num1[i] - '0';
            }
            if (j >= 0) {
                carry += num2[j] - '0';
            }
            //当前为的最大值不大于10
            result += (char)(carry % 10 + '0');
            //如果大于10,向上进一位
            carry /= 10;
            i--;
            j--;
        }
        //相加完之后，如果还有进位，则再加1
        if (carry == 1) {
            result += '1';
        }
        //整体逆置
        reverse(result.begin(), result.end());
        return result;
    }

int main()
{
    string s1, s2;
```

```
    while(cin>>s1>>s2)
    {
        cout<<addStrings(s1, s2)<<endl;
    }
    return 0;
}
```