

青少年软件编程（C 语言）等级考试试卷（三级）

分数：100 题数：5

一、编程题(共 5 题，共 100 分)

1. 和数

给定一个正整数序列，判断其中有多少个数，等于数列中其他两个数的和。比如，对于数列 1 2 3 4, 这个问题的答案就是 2, 因为 $3 = 2 + 1$, $4 = 1 + 3$ 。

时间限制：10000

内存限制：65536

输入

共两行，第一行是数列中数的个数 n ($1 \leq n \leq 100$)，第二行是由 n 个不大于 10000 的正整数组成的数列，相邻两个整数之间用单个空格隔开。

输出

一个整数，即数列中等于其他两个数之和的数的个数。

样例输入

```
4
1 2 3 4
```

样例输出

```
2
```

试题编号：20230318-3-01

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：20

是否评分：已评分

评价描述：

考生答案：

```
#include <iostream>
```

```
#include <cmath>
```

```
#include <fstream>
```

```
#include <sstream>
```

```
#include <vector>
```

```
#include <array>
```

```

#include <string>
#include <algorithm>
#include <cstring>

using namespace std;

int main() {
    int N[100];
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cin >> N[i];
    }

    int total = 0;
    for (int c = 0; c < n; c++) {
        bool done = false;
        for (int a = 0; a < n && !done; a++) {
            for (int b = a; b < n && !done; b++) {
                if (a != b && a != c && b != c && N[c] == N[a] + N[b]) {
                    total++;
                    done = true;
                }
            }
        }
    }

    cout << total << endl;
    return 0;
}

```

2. 质数的和与积

两个质数的和是 S ，它们的积最大是多少？

时间限制：10000

内存限制：65536

输入

一个不大于 10000 的正整数 S ，为两个质数的和。

输出

一个整数，为两个质数的最大乘积。数据保证有解。

样例输入

50

样例输出

589

试题编号：20230318-3-02

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：20

是否评分：已评分

评价描述：

考生答案：

```
#include <iostream>
using namespace std;
bool ss(int x)
{
    for(int i=2;i<x;i++)
    {
        if(x%i==0)
        {
            return false;
            break;
        }
    }
    return true;
}
int main()
{
    int x;
    cin>>x;

    for(int i=x/2;i>=1;i--)
    {
        if(ss(i)&&ss(x-i))
        {
            cout<<i*(x-i);
```

```
        break;
    }
}
return 0;
}
```

3. 爬楼

已知楼梯的数量，可以每次走 2 级或者 3 级，求不同的走法数

例如：楼梯一共有 7 级，一共 3 种方法：2 2 3 或者 2 3 2 或者 3 2 2。

时间限制：1000

内存限制：65536

输入

输入包含若干行，每行包含一个正整数 N，代表楼梯级数， $1 \leq N \leq 50$ 。最后一行为 0，表示测试结束。

输出

不同的走法数，每一行输入对应一行输出

样例输入

```
7
0
```

样例输出

```
3
```

试题编号：20230318-3-03

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：0

是否评分：已评分

评价描述：

4. 生成括号

Paul 是一名数学专业的同学，在课余选修了 C++ 编程课，现在他能够自己写程序判断判断一个给定的由 '(' 和 ')' 组成的字符串是否是正确匹配的。可是他不满足于此，想反其道而行之，设计一个程序，能够生成所有合法的括号组合，请你帮助他解决这个问题。

时间限制：1000
内存限制：65536

输入

输入只有一行 N，代表生成括号的对数 ($1 \leq N \leq 10$)。

输出

输出所有可能的并且有效的括号组合，按照字典序进行排列，每个组合占一行。

样例输入

3

样例输出

((()))
(()())
()(())
()()()
()()()

试题编号：20230318-3-04

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：[点击浏览](#)

考生答案：（此题已作答）

考生得分：20

是否评分：已评分

评价描述：

考生答案：

```
#include<iostream>
#include<bitset>
#include<ctime>
using namespace std;
int N;
bool judge(bitset<20> temp){
    int numL=0,numR=0;
    for(int i=2*N-1;i>=0;i--){
        if(temp[i])numR++;
        else numL++;
    }
}
```

```

        if(numL-numR<0)return false;
    }
    if(numL!=numR)return false;
    return true;
}
int main(){
    cin>>N;
    int max = 1<<(2*N-1);
    for(int i=1;i<=max;i++){
        bitset<20> temp;
        for(int j=0;j<2*N;j++){
            temp[j]= (i>>j)&1;
        }
        if(judge(temp)){
            for(int j=2*N-1;j>=0;j--){
                if(temp[j])cout<<" ";
                else cout<<" ";
            }
            cout<<endl;
        }
    }

    return 0;
}

```

5. 铺砖

对于一个 2 行 N 列的走道。现在用 1*2,2*2 的砖去铺满。问有多少种不同的方式。

时间限制：3000

内存限制：131072

输入

整个测试有多组数据，请做到文件底结束。每行给出一个数字 N， $0 \leq n \leq 250$

输出

如题

样例输入

```
2
8
12
100
200
```

样例输出

```
3
171
2731
845100400152152934331135470251
1071292029505993517027974728227441735014801995855195223
534251
```

试题编号：20230318-3-05

试题类型：编程题

标准答案：

试题难度：一般

试题解析：

展示地址：点击浏览

考生答案：（此题已作答）

考生得分：20

是否评分：已评分

评价描述：

```
考生答案：#include<iostream>
#include <algorithm>
using namespace std;
```

```
string myadd(string a,string b){
    reverse(a.begin(),a.end());
    reverse(b.begin(),b.end());
    int len;
    if(a.length()>b.length()){
        len=a.length();
```

```

        while(b.length()==len) b+=" ";
    }else{
        len=b.length();
        while(a.length()==len) a+=" ";
    }
    int i=0,t=0;
    string ans;
    while(i<len){
        t+=a[i]+b[i]-'0'-'0';
        ans+=(t%10+'0');
        t/=10;
        i++;
    }
    if(t>0) ans+=t+'0';
    reverse(ans.begin(),ans.end());
    return ans;
}

int main(){
    int n;
    string dp[300];
    dp[0]="1";
    dp[1]="1";
    dp[2]="3";
    for(int i=3;i<=250;i++){
        dp[i]=myadd(dp[i-1],myadd(dp[i-2],dp[i-3]));
    }
    while(cin>>n){
        cout << dp[n] <<endl;
    }
    return 0;
}

```