

算法作业1

目录：

- 算法作业1
 - 1.Polycarp and Coins
 - 题目描述
 - 题目分析
 - 解题思路
 - 代码实现
 - 2.Wonderful Coloring - 1
 - 题目描述
 - 题目分析
 - 解题思路
 - 代码实现
 - 3.数楼梯
 - 题目描述
 - 题目分析
 - 解题思路
 - 代码实现
 - 4.蜜蜂路线
 - 题目描述
 - 题目分析
 - 解题思路
 - 代码实现
 - 5. [NOIP2002 普及组] 选数
 - 题目描述
 - 题目分析
 - 解题思路
 - 代码实现

1.Polycarp and Coins

(来源：codeforces)

题目链接：<https://codeforces.ml/contest/1551/problem/A>

题目描述

Polycarp must pay exactly n burles at the checkout. He has coins of two nominal values: 1 burle and 2 burles. Polycarp likes both kinds of coins equally. So he doesn't want to pay with more coins of one type than with the other.

Thus, Polycarp wants to minimize the difference between the count of coins of 1 burle and 2 burles being used. Help him by determining two non-negative integer values c_1 and c_2 which are the number of coins of 1 burle and 2 burles, respectively, so that the total value of that number of coins is exactly n (*i. e.* $C_1 + 2 \cdot C_2 = n$), and the absolute value of the difference between C_1 and C_2 is as little as possible (*i. e.* you must minimize $|C_1 - C_2|$).

题目翻译：

Polycarp必须完全支付 n 结账时的毛病。他有两个面值的硬币：1burle和2burles。Polycarp同样喜欢这两种硬币。所以他不想用一种类型的硬币支付比另一种多。

因此，Polycarp想要最小化硬币数量之间的差异1burle和2burle。通过确定两个非负整数值来帮助他 C_1 和 C_2 这是硬币的数量1burle和2burles，因此该数量的硬币的总价值正好是 n (*i. e.* $C_1 + 2 \cdot C_2 = n$)，以及两者之间差值的绝对值 C_1 和 C_2 尽可能少（即您必须最小化 $|C_1 - C_2|$ ）。

题目分析

给定两个面值不同的硬币，求花费一定数目钱使得花费的两种硬币的数量之差达到最小时两种硬币的数量。

解题思路

先给两块钱，剩下的钱就是一块钱的，因为除法可能不平衡，在把差值抹掉平衡就好。

代码实现

```

1 | #include<bits/stdc++.h>
2 | using namespace std;
3 | int n,m,l,r;
4 | int main(){
5 |     ios::sync_with_stdio(false);
6 |     cin.tie(0);
7 |     cin>>n;
8 |     for(int i=0;i<n;i++){
9 |         cin>>m;
10 |         l=m/3*2,r=m-l,l/=2;
11 |         while(r-l>=2)r-=2,l++;
12 |         cout<<r<<" "<<l<<endl;
13 |     }
14 |     return 0;
15 | }

```

2.Wonderful Coloring - 1

(来源: codeforces)

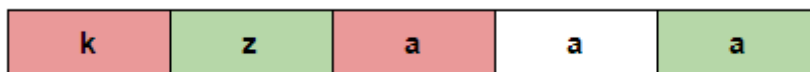
题目链接: <https://codeforces.ml/contest/1551/problem/B1>

题目描述

Paul and Mary have a favorite string s which consists of lowercase letters of the Latin alphabet. They want to paint it using pieces of chalk of two colors: red and green. Let's call a coloring of a string wonderful if the following conditions are met:

1. each letter of the string is either painted in exactly one color (red or green) or isn't painted;
2. each two letters which are painted in the same color are different;
3. the number of letters painted in red is equal to the number of letters painted in green;
4. the number of painted letters of this coloring is maximum among all colorings of the string which meet the first three conditions.

E. g. consider a string s equal to "kzaaa". One of the wonderful colorings of the string is shown in the figure.



The example of a wonderful coloring of the string "kzaaa". Paul and Mary want to learn by themselves how to find a wonderful coloring of the string. But they are very young, so they need a hint. Help them find k — the number of red (or green, these numbers are equal) letters in a wonderful coloring.

题目翻译：

保罗和玛丽有一根最喜欢的弦 秒它由拉丁字母的小写字母组成。他们想用两种颜色的粉笔来画它：红色和绿色。如果满足以下条件，让我们将字符串的着色称为美妙的：

- 1.字符串的每个字母要么只涂上一种颜色（红色或绿色），要么不涂；
- 2.涂上相同颜色的两个字母各不相同；
- 3.涂红色字母的数量等于涂绿色字母的数量；
- 4.在满足前三个条件的字符串的所有颜色中，该颜色的涂色字母数量最多。

例如 考虑一个字符串秒等于“kzaaa”。弦的美妙色彩之一如图所示。

k	z	a	a	a
---	---	---	---	---

字符串“kzaaa”的美妙着色示例。保罗和玛丽想自己学习如何找到美妙的弦颜色。但他们还很年轻，所以他们需要一个提示。帮助他们找到克 — 美妙色彩的红色（或绿色，这些数字是相等的）字母的数量。

题目分析

输入字符串，对其涂色，且满足以下条件：

- 1：一个字符可以被红绿涂色或者不涂。
- 2：一种字符不能被一种颜色涂两次。
- 3：红绿字符数量相同。
- 4：满足前三个条件，尽可能多地涂色。

有多组样例，求出每组被涂色为红的字符数。

解题思路

一个字符出现再多次也只可能被红和绿各涂一次，其余都是空白，开一个数组计数器（a[1010]）和一个数字计数器（记录可以涂色的字符个数，用m表示，初始为0），代码中记录每个字符的出现次数，如果小于2就让m加1，但是m是涂色总数，题目问的是红色数量，红色和绿色数量是一样的，只需m除以二就是最终答案。

注意：每次的统计都要初始化a数组和计数器m:初始化a数组：`cpp memset(a,0,sizeof(a))`

代码实现

```

1  #include<bits/stdc++.h>
2  using namespace std;
3  string s;
4  int n,m,k,a[1010];
5  int main(){
6      ios::sync_with_stdio(false);
7      cin.tie(0);
8      cin>>n;
9      for(int i=0;i<n;i++){
10         memset(a,0,sizeof(a));
11         cin>>s;m=0;
12         for(int i=0;i<s.size();i++){
13             if(a[s[i]-'0']<2) m++;
14             a[s[i]-'0']++;
15         }
16         cout<<m/2<<endl;
17     }
18     return 0;
19 }

```

3.数楼梯

(来源：洛谷 - P1255)

题目链接： <https://www.luogu.com.cn/problem/P1255>

题目描述

楼梯有 N 阶，上楼可以一步上一阶，也可以一步上二阶。

编一个程序，计算共有多少种不同的走法。

题目分析

编一个程序，计算共有多少种不同的走法。

解题思路

当 $n = 1$ 时，有1种； $n = 2$ 时，有2种； $n = 3$ 时，有3种； $n = 4$ 时，有5种； $n = 5$ 时，有8种……； $n = n$ 时，有 $n - 1$ 时的种数 + $n - 2$ 时的种数。是斐波那契数列。需要注意的是， $n \leq 5000$ 求出来的数非常大所以需要对求和进行高精度计算。高精度算法规则：先计算第一位数，然后和下一个数的对等位数相加，如果大于10，则将对10取余后的数留下（储存到数组的当前位），将对10做除法的数进到下一位（储存到数组的下一位）；如果小于10则直接储存到数组的当前位。一直从1算到 n ，最后倒序输出数组的元素即可。

代码实现

```
1 #include<bits/stdc++.h>
2 using namespace std;
3 int n,m=1,a[5005][5005];
4 void hpa(int x){
5     for(int j=1;j<=m;j++) a[x][j]=a[x-1][j]+a[x-2][j];
6     for(int j=1;j<=m;j++) {
7         if(a[x][j]>=10){
8             a[x][j+1]+=a[x][j]/10;
9             a[x][j]=a[x][j]%10;
10            if(a[x][m+1]) m++;
11        }
12    }
13 }
14 int main(){
15     cin>>n;
16     a[1][1]=1;a[2][1]=2;
17     for(int i=3;i<=n;i++) hpa(i);
18     for(int i=m;i>=1;i--) cout<<a[n][i];
19     return 0;
20 }
```

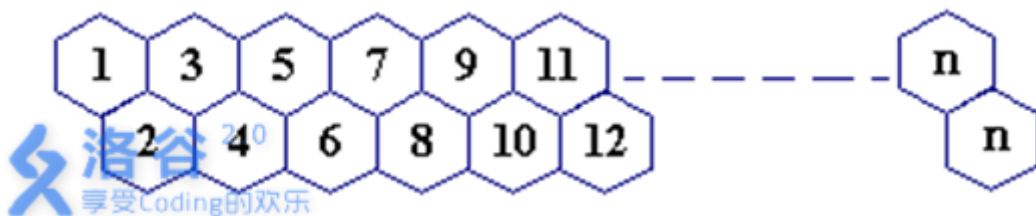
4.蜜蜂路线

(来源：洛谷 - P2437)

题目链接：<https://www.luogu.com.cn/problem/P2437>

题目描述

一只蜜蜂在下图所示的数字蜂房上爬动,已知它只能从标号小的蜂房爬到标号大的相邻蜂房,现在问你:蜜蜂从蜂房 m 开始爬到蜂房 n , $m < n$, 有多少种爬行路线? (备注: 题面有误, 右上角应为 $n - 1$)



题目分析

该题与上题的分析一致, 属于高精斐波那契数列问题

解题思路

高精斐波那契数列，只是需要计算到第(终点一起点)的斐波那契数的值得问题，思路与上题基本一致。

代码实现

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  int n,k,m=1,a[5005][5005];
4  void hpa(int x){
5      for(int j=1;j<=m;j++) a[x][j]=a[x-1][j]+a[x-2][j];
6      for(int j=1;j<=m;j++) {
7          if(a[x][j]>=10){
8              a[x][j+1]+=a[x][j]/10;
9              a[x][j]=a[x][j]%10;
10             if(a[x][m+1]) m++;
11         }
12     }
13 }
14 int main(){
15     cin>>n>>k;
16     a[1][1]=1;a[2][1]=2;
17     for(int i=3;i<=k-n;i++) hpa(i);
18     for(int i=m;i>=1;i--) cout<<a[k-n][i];
19     return 0;
20 }
```

5. [NOIP2002 普及组] 选数

(来源：洛谷 - P1036)

题目链接：<https://www.luogu.com.cn/problem/P1036>

题目描述

已知 n 个整数 x_1, x_2, \dots, x_n ，以及1个整数 $k(k < n)$ 。从 n 个整数中任选 k 个整数相加，可分别得到一系列的 $和$ 。例如当 $n = 4, k = 3$ ，4个整数分别为3, 7, 12, 19时，可得全部的组合与它们的 $和$ 为：

$$3 + 7 + 12 = 22$$

$$3 + 7 + 19 = 29$$

$$7 + 12 + 19 = 38$$

$$3 + 12 + 19 = 34。$$

现在，要求你计算出和为素数共有多少种。

例如上例，只有一种的和为素数： $3 + 7 + 19 = 29$ 。

题目分析

在n个数中选出k个数求和，和为素数的个数。

解题思路

先对整体求和，在用线性筛筛选到最大素数；然后对n个数进行深搜，枚举出在n个数中选出k个的不同情况（规则：从第一个开始加和，在列举下一个进行相加，最后当个数达到k时进行判断；然后再从第二个数…），然后对各种情况进行加和，判断是否为素数。

代码实现

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  typedef long long ll;
4  bool prime[100000010];
5  int n,m,k,l,c,a[1010];
6  ll maxn;
7  void pri(ll end){
8      prime[0]=prime[1]=1;
9      for(ll i=2;i*i<=end;i++){
10         if(!prime[i]){
11             for(ll j=i*i;j<=end;j+=i) prime[j]=1;
12         }
13     }
14 }
15 void dfs(int r,int l,int num){
16     if(l==k){if(!prime[num]){c++;}return;}
17     for(int i=r;i<=n;i++) dfs(i+1,l+1,num+a[i]);
18 }
19 int main(){
20     ios::sync_with_stdio(false);cin.tie(0);
21     cin>>n>>k;
22     for(int i=1;i<=n;i++)cin>>a[i],maxn+=a[i];
23     pri(maxn); dfs(1,0,0);
24     cout<<c<<endl;
25     return 0;
26 }
```