**2014预赛第三题**

标题：李白打酒

话说大诗人李白，一生好饮。幸好他从不开车。

一天，他提着酒壶，从家里出来，酒壶中有酒2斗。他边走边唱：

无事街上走，提壶去打酒。

逢店加一倍，遇花喝一斗。

这一路上，他一共遇到店5次，遇到花10次，已知最后一次遇到的是花，他正好把酒喝光了。

请你计算李白遇到店和花的次序，可以把遇店记为a，遇花记为b。则：babaabbabbabbbb 就是合理的次序。像这样的答案一共有多少呢？请你计算出所有可能方案的个数（包含题目给出的）。

注意：通过浏览器提交答案。答案是个整数。不要书写任何多余的内容。

#include <cstdio>

using namespace std;

int sum = 0;

char str[100];

int Fun(int now, int i, int a, int b)

{

if(now < 0 || i > 16 || (now == 0 && i < 16))

return 0;

if(now == 0)

{

if(i == 16 && a == 5 && b == 10)

{

sum++;

for(int j = 0; j < 15; j++)

putchar(str[j]);

putchar(10);

}

}

str[i - 1] = 'a';

Fun(now \* 2, i + 1, a + 1, b);

str[i - 1] = 'b';

Fun(now - 1, i + 1, a, b + 1);

}

int main()

{

str[15] = '\0';

Fun(2, 1, 0, 0);

printf("sum = %d\n", sum);

return 0;

}

**2013预赛3: 第39级台阶（满分8分）**

小明刚刚看完电影《第39级台阶》，离开电影院的时候，他数了数礼堂前的台阶数，恰好是39级!

站在台阶前，他突然又想着一个问题：

如果我每一步只能迈上1个或2个台阶。先迈左脚，然后左右交替，最后一步是迈右脚，也就是说一共要走偶数步。那么，上完39级台阶，有多少种不同的上法呢？

请你利用计算机的优势，帮助小明寻找答案。

解答：

/有左右脚的限制，即第一步必须左脚，然后左右交替，最后一步必须是右脚。即必须走偶数步。

#include<iostream.h>

//有左右脚的限制。

const int N=39;

int f(int m,int n)

{

if(m==0||n==0)

return 1;

return(f(m-1,n)+f(m,n-1));//递归的关键在此，大规模逐渐转化为小规模。

}

int main()

{

int x=N/2,y; //x表示走两步的次数，y表示走一步的次数。

int i,sum=0;

for(i=x;x>=0;x-=2) //为了保持偶数步，必须x每次递减2，而不是1；(x要x>=0,不能x>0)，x=0是针对偶数台阶。

{

y=N-2\*x;

sum+=f(x,y); //求组合数;

}

cout<<"共有 "<<sum<<"种走法。"<<endl;

return 0;

}

//51167078种走法。

**2012预赛4：**

某电视台举办了低碳生活大奖赛。题目的计分规则相当奇怪：  
 每位选手需要回答10个问题（其编号为1到10），越后面越有难度。答对的，当前分数翻倍；答错了则扣掉与题号相同的分数（选手必须回答问题，不回答按错误处理）。  
  
 每位选手都有一个起步的分数为10分。  
 某获胜选手最终得分刚好是100分，如果不让你看比赛过程，你能推断出他（她）哪个题目答对了，哪个题目答错了吗？  
 如果把答对的记为1，答错的记为0，则10个题目的回答情况可以用仅含有1和0的串来表示。例如：0010110011 就是可能的情况。   
 你的任务是算出所有可能情况。每个答案占一行。

**2011预赛9. 程序设计（满分16分）**

公司发了某商店的购物券1000元，限定只能购买店中的m种商品。每种商品的价格分别为m1,m2,…，要求程序列出所有的正好能消费完该购物券的不同购物方法。

程序输入：

第一行是一个整数m，代表可购买的商品的种类数。

接下来是m个整数，每个1行，分别代表这m种商品的单价。

程序输出：

第一行是一个整数，表示共有多少种方案

第二行开始，每种方案占1行，表示对每种商品购买的数量，中间用空格分隔。

例如：

输入：

2

200

300

则应输出：

2

2  2

5  0

输入：

2

500

800

则应输出：

1

2  0

//预赛NO.9题  
#include <stdio.h>  
  
int sln;//方案的个数  
int gm;//商品的种类  
int price[1000];//各种商品价钱  
int count[1000];//各种商品的个数  
int method[1000][1000];//每种解决方案中各商品的个数  
int cost;//当前花费  
void output()//输出解决方案  
{  
 int i,j;  
 printf("%d\n",sln);  
 for (i=0; i<sln; i++)  
 {  
 for (j=0;j<gm;j++)  
 printf("%3d",method[i][j]);  
 printf("\n");  
 }  
}  
void fun(int m)  
{  
 int i;  
 if (cost == 1000)  
 {  
 for (i=0; i<gm ; i++)  
 method[sln][i] = count[i];  
 sln++;  
 return;  
 }  
 if (cost>1000 || m<0)  
 return;  
 //choose m  
 ++count[m];  
 cost += price[m];  
 fun(m);  
 //not choose m  
 --count[m];  
 cost -= price[m];  
 fun(m-1);  
}  
void main()  
{  
 int m;//商品的种类  
 int k;  
  
 scanf("%d",&m);  
 gm=m;  
 for (k=0;k<m;k++) scanf("%d",&price[k]);  
 fun(m-1);   
 output();  
}