递推迭代

姓名____________

编程解决以下问题

1、 上楼梯

设有一个 N 级的楼梯（4 ≤ N ≤12），编号从下到上依次为 1 至 N。有一个人上楼梯时一步可走一级、或 2 级、或 3 级。问：这个人从楼下走到第 N 级，共有多少种不同的走法？例如：

N=1 时，仅有一种走法；

N=2 时，有 1 级+1 级或 2 级，共 2 种走法

N=3 时，有：1 级+1 级+1 级，或 1 级+2 级，或 2 级+1 级，或 3 级，共 4 种走法。

【程序说明】

用递推方法求解。 {“回头望月”找三个子问题,再利用加法原理相加即可}

【程序清单一】

```
#include <cstdio>
#include <iostream>
using namespace std;

int n;

int main()
{
   cin >> n;
   int f[n+1];

   f[1]=1;  f[2]=2; f[3]=4;

   for (int i=4; i<=n; i++)
      f[i] = f[i-3] + f[i-2] + f[i-1];

   cout << f[n];
   return 0;
}
```

2、 求数组元素

已知给出任意一个自然数(N<=100)，输出满足下列条件的数组元素及不同方案数，条件是：

1) 数组元素由各不相同的自然数组成；
2) 最后一个元素必为 N；
3) 每一个元素都不小于它前面一个元素的平方（第一个元素除外）；
4) 数组中包含的元素个数可以不相同，但至少要有一个元素；

例如：

输入：N=1

输出：数组: (1)

K=1 ‘用 K 记录不同的方案数

又如：

N=5

数组： (5)，(1，2，5)，(1，5)， (2,5)

K=4

【问题分析】

求数组元素的不同方案，其实是一个带有约定条件的排列组合问题。从问题的描述中可以知道，要找到后一个元素不小于前一个元素的平方，而且最后一个元素必须为 N，换句话说，就是要找小于 N 的自然数组，其中元素的平方都小于 N，而且满足问题描述中的条件。

例如，当 N=9 时，其平方小于等于 9 的数有 1，2，3，其中 $1^2<2$，$1^2<3$，但 $2^2>3$，所以数组中 2、3 两个数不能同时出现。

详细方案如下：

N=9

方案：

(9)，(3,9)，(1,3,9)，(2,9)，(1,2,9)，(1,9)

共有 6 种不同的方案。

下面，再罗列出 N 取不同值时的方案，然后从中找出规律：

N=1	(1)	K(1)=1
N=2	(2),(1,2)	K(2)=2
N=3	(3),(1,3)	K(3)=2
N=4	(4),(2,4),(1,2,4),(1,4)	K(4)=4
N=5	(5),(2,5),(1,2,5),(1,5)	K(5)=4
N=6	(6),(2,6),(1,2,6),(1,6)	K(6)=4
……	……	……
N=9	(9),(3,9),(1,3,9),(1,2,9),(2,9),(1,9)	K(9)=6

可以看出：

K(2) = K(1) + 1

K(3) = K(1) + 1

K(4) = K(2) + K(1) + 1

K(5) = K(2) + K(1) + 1

……

K(9) = K(3) + K(2) + K(1) + 1

由此，总结出递推公式：K(N) = K(M) + K(M-1) + … + K(1) + 1

其中，m = int(sqrt(n))

【算法的主要步骤】

S(1)：先利用双重循环将 10 以内的方案数计算后，存放在数组 A 中；

S(2)：输入 N；

S(3)：在一重循环中，利用公式直接求和；

S(4)：输出方案数 K。

【参考程序】

```
#include <cstdio>
#include <iostream>
#include <cmath>
using namespace std;

int n;

int main()
{
   cin >> n;
   int f[n+1];

   f[1]=1;
   f[n]+=1+f[1];

   for (int i=2; i<=int(sqrt(n)); i++)
   {
      f[i]=1;
      for (int j=1; j<=int(sqrt(i)); j++)
         f[i] =f[i] + f[j];

      f[n]+=f[i];
   }

   cout << f[n];
   return 0;
}
```

3、 **数的计数**（count.bas）

【问题描述】

要求找出具有下列性质的数的个数(包含输入的自然数 n)：

先输入一个自然数 n （n≤1000)，然后对此自然数按照如下方法进行处理：

1. 不作任何处理；
2. 在它的左边加上一个自然数，但该自然数不能超过原数的一半；
3. 加上数后，继续按此规则进行处理，直到不能再加自然数为止。

【输入输出样例】

输入：6

输出：6

【输出样例说明】

满足条件的数为（此部分不必输出）：

6

16

26

126

36

136

【问题分析】

F(6)=F(1)+ F(2)+ F(3)+1

【参考程序】

```
#include <cstdio>
#include <iostream>
using namespace std;

int i, j, n;

int main()
{
   cin >> n;
   int f[n+1];

   f[1]=1;
   for (i=2; i<=n; i++)
   {
      f[i]=1;
      for (j=1; j<= i/2; j++)
         f[i] += f[j];
   }

   cout << f[n];
   return 0;
}
```

4、 回文数列 {回溯; 或者:从 1111 向下穷举,找出所有的回文数,看其数位和是否等于 4,且不能含 0}

【问题描述】

对一个正整数 K，求出 K 的所有拆分，并统计输出其中回文数列的个数。所谓回文数列是指该数列中的所有数字，从左向右或从右向左看都相同。例如，K=4 时，有如下的拆分：

4 = 1 + 1 + 1 + 1 {回文数列 1}

= 1 + 1 + 2

= 1 + 2 + 1　　　　{回文数列 2}

= 2 + 1 + 1

= 2 + 2　　　　　{回文数列 3}

= 1 + 3

= 3 + 1

因此，回文数列共有 3 个。

【输入】

一个正整数 K（1<K≤26）。

【输出】

满足条件的回文数列的个数。

【输入样例】

4

【输出样例】

3

【分析】

经归纳总结发现，满足规律：$2^{k\backslash 2}-1$，分析如下：

以 K=4 为例说明：

K=1+1+1+…+1　　　　'共 K 个 1，K-1 个加号

K=1 | 1 | 1 |…| 1　　　'将 K-1 个加号看作是 K-1 个用于分隔的"|"线

'每条分隔线可以取或不取，代表拆分或不拆分

要想形成回文拆分，则以中间那条分隔线为中心，两边对称的分隔线应同时出现或同时不出现，亦即，问题相当于/转换化对分隔线做 0-1 穷举；

由于两边的分隔线的穷举是对称的，则只要穷举一半的分隔线即可，则为：

2^(((k-1)+1)\2) - 1 = 2^(k\2) - 1

其中，最后减去 1 是因为"全 0"的穷举是不行的——单独一个 4 是不算的。

【问题】

如何产生回文数列？

5、极值问题

【问题描述】

已知 m、n 为整数，且满足下列两个条件：

1)m、n∈{1，2，…，k}，即 1<=m,n<=k;

2)$(n^2-mn-m^2)^2=1$

你的任务是：编程由键盘输入正整数 k（$1<=k<=10^9$），求一组满足上述两个条件的 m、n，并且使 m^2+n^2 的值最大。例如，从键盘输入 k=1995，则输出：m=987，n=1597。

【参考程序】

```
#include <cstdio>
#include <iostream>
using namespace std;

long long k, a, b, c;
```

```
int main()
{
   cin >> k;
   a = 1;  b = 1;
   do {
      c = a+b;
      if (c>k) break;
      a = b;
      b = c;
   } while (1);

   cout << a << ' ' << b;
   return 0;
}
```