**整数划分问题**

整数划分是一个经典的问题。希望这道题会对你的组合数学的解题能力有所帮助。   
  
Input   
  
每组输入是两个整数n和k。(1 <= n <= 50, 1 <= k <= n)   
  
Output   
  
对于每组输入，请输出六行。   
  
第一行： 将n划分成若干正整数之和的划分数。   
第二行： 将n划分成k个正整数之和的划分数。   
第三行： 将n划分成最大数不超过k的划分数。   
第四行： 将n划分成若干奇正整数之和的划分数。   
第五行： 将n划分成若干不同整数之和的划分数。   
第六行： 打印一个空行。   
  
Sample Input   
5 2  
  
Sample Output   
7  
2  
3  
3  
3  
  
  
Hint:   
1、将5划分成若干正整数之和的划分为： 5, 4+1, 3+2, 3+1+1, 2+2+1, 2+1+1+1, 1+1+1+1+1   
2、将5划分成2个正整数之和的划分为： 3+2, 4+1   
3、将5划分成最大数不超过2的划分为： 1+1+1+1+1, 1+1+1+2, 1+2+2   
4、将5划分成若干奇正整数之和的划分为： 5, 1+1+3, 1+1+1+1+1   
5、将5划分成若干不同整数之和的划分为： 5, 1+4, 2+3

来源：[http://acm.hit.edu.cn/ojs/show.php?Proid=1402&Contestid=0](http://acm.hit.edu.cn/ojs/show.php?Proid=1402&Contestid=0" \t "_blank)

**最新评论** [**发表评论**](http://www.programfan.com/acm/show.asp?qid=57#addcomment)

**您尚未登陆本站，不能发表评论，请[登陆](http://www.programfan.com/logon.asp?fromurl=%2Facm%2Fshow%2Easp%3Fqid%3D57)**

评论人: KnightZLJ  发布时间: 2009-11-5 0:32:44

#include<iostream.h>  
int Func\_a(int n,int max)            // 将n划分成最大数不超过max的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;i<=max;i++)  
        {  
            counter+=Func\_a(n-i,(i<n-i)?i:(n-i));  
        }  
  
        return counter;  
    }  
}  
  
int Func\_b(int n,int k,int max)            //将n划分成k个不大于max的正整数之和的划分数。  
{  
    int counter=0;  
    int min,i;  
  
    min=(int)((n+k-1)/k);  
    if(k==1||k==n)  
    {  
        return 1;  
    }  
    else  
    {  
        min=n-(int)(n/k)\*(k-1);  
        for(i=min;(i<=n-k+1&&i<=max);i++)  
        {  
  
            counter+=Func\_b(n-i,k-1,i);  
        }  
  
        return counter;  
    }  
}  
  
int Func\_c(int n,int max)            //将n划分成若干奇正整数之和的划分数。  
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;2\*i<=max+1;i++)  
        {  
            counter+=Func\_c(n-2\*i+1,(i<n-2\*i+1)?i:(n-2\*i+1));  
        }  
        return counter;  
    }  
}  
  
  
  
  
int Func\_d(int n,int max)            // 将n划分成若干不同整数之和的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max\*(max-1)/2<n)  
    {  
        return 0;  
    }else{  
        if(max\*(max-1)/2==n||n==0)  
        {  
            return 1;  
        }else{  
                for(i=1;i<max;i++)  
                {  
                    counter+=Func\_d(n-i,i);  
                }  
                    return counter;  
        }  
    }  
}  
  
  
  
  
void main()  
{  
    int n,k;  
    cout<<"输入是两个整数n和k。(1 <= n <= 50, 1 <= k <= n)\n";  
    cin>>n>>k;  
    cout<<Func\_a(n,n)<<endl;//第一行： 将n划分成若干正整数之和的划分数。   
  
    cout<<Func\_b(n,k,n-k+1)<<endl;//第二行： 将n划分成k个正整数之和的划分数。   
  
    cout<<Func\_a(n,k)<<endl;//第三行： 将n划分成最大数不超过k的划分数。   
  
  
    cout<<Func\_c(n,n)<<endl;//第四行： 将n划分成若干奇正整数之和的划分数。   
  
    cout<<Func\_d(n,n)<<endl;//第五行： 将n划分成若干不同整数之和的划分数。   
  
    cout<<endl;//第六行： 打印一个空行。   
  
}

评论人: KnightZLJ  发布时间: 2009-11-5 0:32:28

#include<iostream.h>  
int Func\_a(int n,int max)            // 将n划分成最大数不超过max的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;i<=max;i++)  
        {  
            counter+=Func\_a(n-i,(i<n-i)?i:(n-i));  
        }  
  
        return counter;  
    }  
}  
  
int Func\_b(int n,int k,int max)            //将n划分成k个不大于max的正整数之和的划分数。  
{  
    int counter=0;  
    int min,i;  
  
    min=(int)((n+k-1)/k);  
    if(k==1||k==n)  
    {  
        return 1;  
    }  
    else  
    {  
        min=n-(int)(n/k)\*(k-1);  
        for(i=min;(i<=n-k+1&&i<=max);i++)  
        {  
  
            counter+=Func\_b(n-i,k-1,i);  
        }  
  
        return counter;  
    }  
}  
  
int Func\_c(int n,int max)            //将n划分成若干奇正整数之和的划分数。  
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;2\*i<=max+1;i++)  
        {  
            counter+=Func\_c(n-2\*i+1,(i<n-2\*i+1)?i:(n-2\*i+1));  
        }  
        return counter;  
    }  
}  
  
  
  
  
int Func\_d(int n,int max)            // 将n划分成若干不同整数之和的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max\*(max-1)/2<n)  
    {  
        return 0;  
    }else{  
        if(max\*(max-1)/2==n||n==0)  
        {  
            return 1;  
        }else{  
                for(i=1;i<max;i++)  
                {  
                    counter+=Func\_d(n-i,i);  
                }  
                    return counter;  
        }  
    }  
}  
  
  
  
  
void main()  
{  
    int n,k;  
    cout<<"输入是两个整数n和k。(1 <= n <= 50, 1 <= k <= n)\n";  
    cin>>n>>k;  
    cout<<Func\_a(n,n)<<endl;//第一行： 将n划分成若干正整数之和的划分数。   
  
    cout<<Func\_b(n,k,n-k+1)<<endl;//第二行： 将n划分成k个正整数之和的划分数。   
  
    cout<<Func\_a(n,k)<<endl;//第三行： 将n划分成最大数不超过k的划分数。   
  
  
    cout<<Func\_c(n,n)<<endl;//第四行： 将n划分成若干奇正整数之和的划分数。   
  
    cout<<Func\_d(n,n)<<endl;//第五行： 将n划分成若干不同整数之和的划分数。   
  
    cout<<endl;//第六行： 打印一个空行。   
  
}

评论人: KnightZLJ  发布时间: 2009-11-5 0:32:13

#include<iostream.h>  
int Func\_a(int n,int max)            // 将n划分成最大数不超过max的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;i<=max;i++)  
        {  
            counter+=Func\_a(n-i,(i<n-i)?i:(n-i));  
        }  
  
        return counter;  
    }  
}  
  
int Func\_b(int n,int k,int max)            //将n划分成k个不大于max的正整数之和的划分数。  
{  
    int counter=0;  
    int min,i;  
  
    min=(int)((n+k-1)/k);  
    if(k==1||k==n)  
    {  
        return 1;  
    }  
    else  
    {  
        min=n-(int)(n/k)\*(k-1);  
        for(i=min;(i<=n-k+1&&i<=max);i++)  
        {  
  
            counter+=Func\_b(n-i,k-1,i);  
        }  
  
        return counter;  
    }  
}  
  
int Func\_c(int n,int max)            //将n划分成若干奇正整数之和的划分数。  
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;2\*i<=max+1;i++)  
        {  
            counter+=Func\_c(n-2\*i+1,(i<n-2\*i+1)?i:(n-2\*i+1));  
        }  
        return counter;  
    }  
}  
  
  
  
  
int Func\_d(int n,int max)            // 将n划分成若干不同整数之和的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max\*(max-1)/2<n)  
    {  
        return 0;  
    }else{  
        if(max\*(max-1)/2==n||n==0)  
        {  
            return 1;  
        }else{  
                for(i=1;i<max;i++)  
                {  
                    counter+=Func\_d(n-i,i);  
                }  
                    return counter;  
        }  
    }  
}  
  
  
  
  
void main()  
{  
    int n,k;  
    cout<<"输入是两个整数n和k。(1 <= n <= 50, 1 <= k <= n)\n";  
    cin>>n>>k;  
    cout<<Func\_a(n,n)<<endl;//第一行： 将n划分成若干正整数之和的划分数。   
  
    cout<<Func\_b(n,k,n-k+1)<<endl;//第二行： 将n划分成k个正整数之和的划分数。   
  
    cout<<Func\_a(n,k)<<endl;//第三行： 将n划分成最大数不超过k的划分数。   
  
  
    cout<<Func\_c(n,n)<<endl;//第四行： 将n划分成若干奇正整数之和的划分数。   
  
    cout<<Func\_d(n,n)<<endl;//第五行： 将n划分成若干不同整数之和的划分数。   
  
    cout<<endl;//第六行： 打印一个空行。   
  
}

评论人: KnightZLJ  发布时间: 2009-11-5 0:31:54

#include<iostream.h>  
int Func\_a(int n,int max)            // 将n划分成最大数不超过max的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;i<=max;i++)  
        {  
            counter+=Func\_a(n-i,(i<n-i)?i:(n-i));  
        }  
  
        return counter;  
    }  
}  
  
int Func\_b(int n,int k,int max)            //将n划分成k个不大于max的正整数之和的划分数。  
{  
    int counter=0;  
    int min,i;  
  
    min=(int)((n+k-1)/k);  
    if(k==1||k==n)  
    {  
        return 1;  
    }  
    else  
    {  
        min=n-(int)(n/k)\*(k-1);  
        for(i=min;(i<=n-k+1&&i<=max);i++)  
        {  
  
            counter+=Func\_b(n-i,k-1,i);  
        }  
  
        return counter;  
    }  
}  
  
int Func\_c(int n,int max)            //将n划分成若干奇正整数之和的划分数。  
{  
    int counter=0;  
    int i;  
  
    if(max<=1)  
    {  
        return 1;  
    }  
    else  
    {  
        for(i=1;2\*i<=max+1;i++)  
        {  
            counter+=Func\_c(n-2\*i+1,(i<n-2\*i+1)?i:(n-2\*i+1));  
        }  
        return counter;  
    }  
}  
  
  
  
  
int Func\_d(int n,int max)            // 将n划分成若干不同整数之和的划分数。   
{  
    int counter=0;  
    int i;  
  
    if(max\*(max-1)/2<n)  
    {  
        return 0;  
    }else{  
        if(max\*(max-1)/2==n||n==0)  
        {  
            return 1;  
        }else{  
                for(i=1;i<max;i++)  
                {  
                    counter+=Func\_d(n-i,i);  
                }  
                    return counter;  
        }  
    }  
}  
  
  
  
  
void main()  
{  
    int n,k;  
    cout<<"输入是两个整数n和k。(1 <= n <= 50, 1 <= k <= n)\n";  
    cin>>n>>k;  
    cout<<Func\_a(n,n)<<endl;//第一行： 将n划分成若干正整数之和的划分数。   
  
    cout<<Func\_b(n,k,n-k+1)<<endl;//第二行： 将n划分成k个正整数之和的划分数。   
  
    cout<<Func\_a(n,k)<<endl;//第三行： 将n划分成最大数不超过k的划分数。   
  
  
    cout<<Func\_c(n,n)<<endl;//第四行： 将n划分成若干奇正整数之和的划分数。   
  
    cout<<Func\_d(n,n)<<endl;//第五行： 将n划分成若干不同整数之和的划分数。   
  
    cout<<endl;//第六行： 打印一个空行。   
  
}

评论人: master\_zcw  发布时间: 2009-10-20 19:37:12

整数划分问题是将一个正整数n拆成一组数连加并等于n的形式，且这组数中的最大加数不大于n。  
    如6的整数划分为  
    最大数    
    6         6  
    5        5 + 1  
    4         4 + 2, 4 + 1 + 1  
    3         3 + 3, 3 + 2 + 1, 3 + 1 + 1 + 1  
    2        2 + 2 + 2, 2 + 2 + 1 + 1, 2 + 1 + 1 + 1 + 1  
    1         1 + 1 + 1 + 1 + 1 + 1  
      
    共11种。下面介绍一种通过递归方法得到一个正整数的划分数。  
      
    递归函数的声明为 int split(int n, int m);其中n为要划分的正整数，m是划分中的最大加数(当m > n时，最大加数为n)，  
    1 当n = 1或m = 1时，split的值为1，可根据上例看出，只有一个划分1 或 1 + 1 + 1 + 1 + 1 + 1  
    可用程序表示为if(n == 1 || m == 1) return 1;  
      
    2 下面看一看m 和 n的关系。它们有三种关系  
    (1) m > n  
    在整数划分中实际上最大加数不能大于n，因此在这种情况可以等价为split(n, n);  
    可用程序表示为if(m > n) return split(n, n);      
    (2) m = n  
    这种情况可用递归表示为split(n, m - 1) + 1，从以上例子中可以看出，就是最大加  
    数为6和小于6的划分之和  
    用程序表示为if(m == n) return (split(n, m - 1) + 1);  
    (3) m < n  
    这是最一般的情况，在划分的大多数时都是这种情况。  
    从上例可以看出，设m = 4，那split(6, 4)的值是最大加数小于4划分数和整数2的划分数的和。  
    因此，split(n, m)可表示为split(n, m - 1) + split(n - m, m)  
      
    根据以上描述，可得源程序如下：  
    
  
#include <stdio.h>  
  
   int split(int n, int m)  
    {  
      if(n < 1 || m < 1) return 0;  
      if(n == 1 || m == 1) return 1;  
      if(n < m) return split(n, n);  
      if(n == m) return (split(n, m - 1) + 1);  
      if(n > m) return (split(n, m - 1) + split((n - m), m));  
   }  
  
int main()  
{  
      printf("12的划分数: %d", split(12, 12));  
    return 0;  
}  
  
将正整数划分成连续的正整数之和  
如15可以划分成4种连续整数相加的形式：  
15  
7 8  
4 5 6  
1 2 3 4 5  
  
    首先考虑一般的形式，设n为被划分的正整数，x为划分后最小的整数，如果n有一种划分，那么  
结果就是x,如果有两种划分，就是x和x x + 1， 如果有m种划分，就是 x 、x x + 1 、 x x + 1 x + 2 、... 、x x + 1 x + 2 ... x + m - 1  
将每一个结果相加得到一个公式(i \* x + i \* (i - 1) / 2) = n，i为当前划分后相加的正整数个数。  
满足条件的划分就是使x为正整数的所有情况。  
如上例，当i = 1时，即划分成一个正整数时，x = 15， 当i = 2时， x = 7。  
当x = 3时，x = 4， 当x = 4时，4/9，不是正整数，因此，15不可能划分成4个正整数相加。  
当x = 5时，x = 1。  
  
    这里还有一个问题，这个i的最大值是多少？不过有一点可以肯定，它一定比n小。我们可以做一个假设，  
假设n可以拆成最小值为1的划分，如上例中的1 2 3 4 5。这是n的最大数目的划分。如果不满足这个假设，  
那么 i 一定比这个划分中的正整数个数小。因此可以得到这样一个公式i \* (i + 1) / 2 <= n，即当i满足  
这个公式时n才可能被划分。  
  
综合上述，源程序如下  
  
int split1(int n)  
{  
    int i, j, m = 0, x, t1, t2;  
   // 在这里i + 1之所以变为i - 1，是因为i \* (i - 1) / 2这个式子在下面多次用到，  
  // 为了避免重复计算，因此将这个值计算完后保存在t1中。并且将<= 号变为了<号。  
    for(i = 1; (t1 = i \* (i - 1) / 2) < n; i++)   
     {  
         t2 = (n - t1);  
         x =   t2 / i;  
        if(x <= 0) break;  
        if((n - t1) % i == 0)  
         {  
             printf("%d ", x);  
            for(j = 1; j < i; j++)  
                 printf("%d ", x + j);  
             printf("\n");  
             m++;  
         }  
     }  
    return m;  
}