**算法训练 星际交流**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　人类终于登上了火星的土地并且见到了神秘的火星人。人类和火星人都无法理解对方的语言，但是我们的科学家发明了一种用数字交流的方法。这种交流方法是这样 的，首先，火星人把一个非常大的数字告诉人类科学家，科学家破解这个数字的含义后，再把一个很小的数字加到这个大数上面，把结果告诉火星人，作为人类的回 答。  
　　火星人用一种非常简单的方式来表示数字——掰手指。火星人只有一只手，但这只手上有成千上万的手指，这些手指排成一列，分别编号为1，2，3……。火星人的任意两根手指都能随意交换位置，他们就是通过这方法计数的。  
　　一个火星人用一个人类的手演示了如何用手指计数。如果把五根手指——拇指、食指、中指、无名指和小指分别编号为1，2，3，4和5，当它们按正常顺序排列 时，形成了5位数12345，当你交换无名指和小指的位置时，会形成5位数12354，当你把五个手指的顺序完全颠倒时，会形成54321，在所有能够形 成的120个5位数中，12345最小，它表示1；12354第二小，它表示2；54321最大，它表示120。下表展示了只有3根手指时能够形成的6个 3位数和它们代表的数字：  
　　三进制数  
　　123  
　　132  
　　213  
　　231  
　　312  
　　321  
　　代表的数字  
　　1  
　　2  
　　3  
　　4  
　　5  
　　6  
　　现在你有幸成为了第一个和火星人交流的地球人。一个火星人会让你看他的手指，科学家会告诉你要加上去的很小的数。你的任务是，把火星人用手指表示的数与科 学家告诉你的数相加，并根据相加的结果改变火星人手指的排列顺序。输入数据保证这个结果不会超出火星人手指能表示的范围。

输入格式

　　包括三行，第一行有一个正整数N，表示火星人手指的数目（1 <= N <= 10000）。第二行是一个正整数M，表示要加上去的小整数（1 <= M <= 100）。下一行是1到N这N个整数的一个排列，用空格隔开，表示火星人手指的排列顺序。

输出格式

　　只有一行，这一行含有N个整数，表示改变后的火星人手指的排列顺序。每两个相邻的数中间用一个空格分开，不能有多余的空格。

样例输入

5  
3  
1 2 3 4 5

样例输出

1 2 4 5 3

数据规模和约定

　　对于30%的数据，N<=15；  
　　对于60%的数据，N<=50；  
　　对于全部的数据，N<=10000；

锦囊1

排列生成。

锦囊2

使用组合公式可以算出某一位增加一个的时候数字是增加多少。从左到右依次考虑，如果增加量还没到要求的就增加。

本题的C++参考代码如下：

#include <iostream>

#include <cmath>

#include <cstring>

#include <algorithm>

#include <cstdio>

using namespace std;

#define du freopen("in.txt","r",stdin)

#define chu freopen("out.txt","w",stdout)

#define FOR(i,a,b) for(int i=a;i<=b;i++)

#define FD(i,a,b) for(int i=a;i>=b;i--)

#define FF(i,a) for(int i=0;i<a;i++)

#define SD(x) scanf("%d",&x)

#define PD(x) printf("%d",x)

#define PP printf(" ")

#define LN puts("")

const int maxn= 0;

int a[20001];

int main()

{

int n;

while(~scanf("%d",&n)){

int m; SD(m);

FF(i,n) SD(a[i]);

while( m&& next\_permutation(a,a+n)){

m--;

}

PD(a[0]);

FOR(i,1,n-1){

PP;PD(a[i]);

}

LN;

}

}

本题的C参考代码如下：

#include<stdio.h>

int main()

{

int n,w,i1,i2;

int a[100000];

scanf("%d%d",&n,&w);

for(i1=0;i1<n;i1++)

scanf("%d",&a[i1]);

while(w--)

{ int max=a[n-1],min;

for(i1=n-2;i1>=0;i1--)

{

if(max<a[i1])

{max=a[i1];

continue;

}

min=max;

int xiabiao=-1;

for(i2=i1+1;i2<=n-1;i2++)

{

if(a[i2]>a[i1])

{

if(min>=a[i2])

{

min=a[i2];

xiabiao=i2;

}

}

}

int k=a[i1];

a[i1]=min;

a[xiabiao]=k;

for(i2=1;i1+i2<n-i2;i2++)

{

int kp=a[i2+i1];

a[i2+i1]=a[n-i2];

a[n-i2]=kp;

}

break;

}

}

for(i1=0;i1<n;i1++)

printf("%d ",a[i1]);

printf("\n");

return 0;

}

本题的Java参考代码如下

import java.io.\*;

public class Main {

public static void main(String args[]) throws NumberFormatException, IOException {

BufferedReader bf=new BufferedReader(new InputStreamReader(System.in));

int n=Integer.parseInt(bf.readLine());

int m=Integer.parseInt(bf.readLine());

String s[]=bf.readLine().split(" ");

int a[] =new int[10001];

int sum=1;

for(int i=1;i<=n;i++)

a[i]=Integer.parseInt(s[i-1]);

for(int i=2;i<=n;i++){

sum=1;

for(int j=i;j>1;j--)

sum\*=j;

int c[]=new int[i+1];

for(int k=0;k<i;k++)

c[k]=a[n-i+k+1];

if(cantor(c,i,m,sum,n,a))

return;

}

}

public static boolean cantor(int a[],int n,int m,int sum,int y,int d[]){

int count = 1;

for (int i = 0; i < n; i++) {

int temp = 0;

for (int j = i + 1; j < n; j++) {

if (a[j] < a[i])

temp++;

}

int num = 1;

for (int k = 2; k <= n - i - 1; k++) {

num \*= k;

}

count += num \* temp;

}

if(count+m<=sum){

//System.out.println(count+" "+sum);

for(int i=1;i<=y-n;i++)

System.out.print(d[i]+" ");

java.util.Arrays.sort(a);

boolean b[]=new boolean[10001];

int r=count+m-1;

sum=1;

int num=0;

for(int i=n;i>=1;i--){

sum=1;

num=0;

for(int j=i-1;j>=1;j--)

sum\*=j;

int x=r/sum;

r=r%sum;

for(int k=1;k<=n;k++){

if(!b[k])

num++;

if(num==x+1){

System.out.print(a[k]+" ");

b[k]=true;

break;

}

}

}

return true;

}

return false;

}

}