BEGIN-4**入门训练 Fibonacci数列**

时间限制：1.0s   内存限制：256.0MB

关键字：入门 数列 取模

问题描述

Fibonacci数列的递推公式为：Fn=Fn-1+Fn-2，其中F1=F2=1。

当n比较大时，Fn也非常大，现在我们想知道，Fn除以10007的余数是多少。

输入格式

输入包含一个整数n。

输出格式

输出一行，包含一个整数，表示Fn除以10007的余数。

说明：在本题中，答案是要求Fn除以10007的余数，因此我们只要能算出这个余数即可，而不需要先计算出Fn的准确值，再将计算的结果除以10007取余数，直接计算余数往往比先算出原数再取余简单。

样例输入

10

样例输出

55

样例输入

22

样例输出

7704

数据规模与约定

1 <= n <= 1,000,000。

**参考代码：**

**C++**

**#include <stdlib.h>**

**#include <stdio.h>**

**#define MOD 10007**

**#define MAXN 1000001**

**int n, i, F[MAXN];**

**int main()**

**{**

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

**F[1] = 1;**

**F[2] = 1;**

**for (i = 3; i <= n; ++i)**

**F[i] = (F[i-1] + F[i-2]) % MOD;**

**printf("%d\n", F[n]);**

**return 0;**

**}**

**C**

**#include <stdlib.h>**

**#include <stdio.h>**

**#define MOD 10007**

**#define MAXN 1000001**

**int n, i, F[MAXN];**

**int main()**

**{**

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

**F[1] = 1;**

**F[2] = 1;**

**for (i = 3; i <= n; ++i)**

**F[i] = (F[i-1] + F[i-2]) % MOD;**

**printf("%d\n", F[n]);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

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

**BufferedReader reader=new BufferedReader(new InputStreamReader(System.in));**

**String s=reader.readLine();**

**int n=Integer.valueOf(s);**

**int f1=1,f2=1,f3=0;**

**if(n<3){**

**System.out.print("1");**

**return;}**

**for(int i=3;i<=n;i++)**

**{if(f1>10007)f1=f1%10007;**

**if(f2>10007)f2=f2%10007;**

**f3=f1+f2;**

**f1=f2;**

**f2=f3;**

**}**

**System.out.print(f3%10007);**

**}**

**}**

BEGIN-3**入门训练 圆的面积**

时间限制：1.0s   内存限制：256.0MB

关键字：入门 实数输出

问题描述

给定圆的半径r，求圆的面积。

输入格式

输入包含一个整数r，表示圆的半径。

输出格式

输出一行，包含一个实数，四舍五入保留小数点后7位，表示圆的面积。

说明：在本题中，输入是一个整数，但是输出是一个实数。

对于实数输出的问题，请一定看清楚实数输出的要求，比如本题中要求保留小数点后7位，则你的程序必须**严格的**输出7位小数，输出过多或者过少的小数位数都是不行的，都会被认为错误。

实数输出的问题如果没有特别说明，舍入都是按四舍五入进行。

样例输入

4

样例输出

50.2654825

数据规模与约定

1 <= r <= 10000。

提示

本题对精度要求较高，请注意π的值应该取较精确的值。你可以使用常量来表示π，比如PI=3.14159265358979323，也可以使用数学公式来求π，比如PI=atan(1.0)\*4。

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <math.h>**

**int main()**

**{**

**int r;**

**double s, PI;**

**scanf("%d", &r);**

**PI = atan(1.0) \* 4;**

**s = PI \* r \* r;**

**printf("%.7lf", s);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <math.h>**

**int main()**

**{**

**int r;**

**double s, PI;**

**scanf("%d", &r);**

**PI = atan(1.0) \* 4;**

**s = PI \* r \* r;**

**printf("%.7lf", s);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.math.BigDecimal;**

**public class Main {**

**/\*\***

**\* @param args**

**\* @throws IOException**

**\*/**

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

**// TODO Auto-generated method stub**

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

**BigDecimal bg = new BigDecimal(bf.readLine());**

**BigDecimal value = bg.multiply(bg.multiply(new BigDecimal(Math.PI))).setScale(7,BigDecimal.ROUND\_HALF\_UP);**

**System.out.println(value);**

**}**

**}**

BEGIN-2**入门训练 序列求和**

时间限制：1.0s   内存限制：256.0MB

关键字：入门 求和公式

问题描述

求1+2+3+...+n的值。

输入格式

输入包括一个整数n。

输出格式

输出一行，包括一个整数，表示1+2+3+...+n的值。

样例输入

4

样例输出

10

样例输入

100

说明：有一些试题会给出多组样例输入输出以帮助你更好的做题。

一般在提交之前所有这些样例都需要测试通过才行，但这不代表这几组样例数据都正确了你的程序就是完全正确的，潜在的错误可能仍然导致你的得分较低。

样例输出

5050

数据规模与约定

1 <= n <= 1,000,000,000。

说明：请注意这里的数据规模。

本题直接的想法是直接使用一个循环来累加，然而，当数据规模很大时，这种“暴力”的方法往往会导致超时。此时你需要想想其他方法。你可以试一试，如果使用1000000000作为你的程序的输入，你的程序是不是能在规定的上面规定的时限内运行出来。

本题另一个要值得注意的地方是答案的大小不在你的语言默认的整型(int)范围内，如果使用整型来保存结果，会导致结果错误。

如果你使用C++或C语言而且准备使用printf输出结果，则你的格式字符串应该写成%I64d以输出long long类型的整数。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**long long n;**

**cin >> n;**

**cout << (1+n) \* n / 2;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int n;**

**long long ret = 0, tmp;**

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

**tmp = n;**

**printf("%I64d", (1+tmp) \* tmp / 2);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

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

**BufferedReader strin = new BufferedReader(new InputStreamReader(System.in));**

**long i = Integer.parseInt(strin.readLine());**

**long sum = (1+i)\*i/2;**

**System.out.println(sum);**

**}**

**}**

BEGIN-1**入门训练 A+B问题**

时间限制：1.0s   内存限制：256.0MB

关键字：入门

问题描述

输入A、B，输出A+B。

说明：在“问题描述”这部分，会给出试题的意思，以及所要求的目标。

输入格式

输入的第一行包括两个整数，由空格分隔，分别表示A、B。

说明：“输入格式”是描述在测试你的程序时，所给的输入一定满足的格式。

做题时你应该假设所给的输入是一定满足输入格式的要求的，所以你不需要对输入的格式进行检查。多余的格式检查可能会适得其反，使用你的程序错误。

在测试的时候，系统会自动将输入数据输入到你的程序中，你不能给任何提示。比如，你在输入的时候提示“请输入A、B”之类的话是不需要的，这些多余的输出会使得你的程序被判定为错误。

输出格式

输出一行，包括一个整数，表示A+B的值。

说明：“输出格式”是要求你的程序在输出结果的时候必须满足的格式。

在输出时，你的程序必须满足这个格式的要求，不能少任何内容，也不能多任何内容。如果你的内容和输出格式要求的不一样，你的程序会被判断为错误，包括你输出了提示信息、中间调试信息、计时或者统计的信息等。

样例输入

12 45

说明：“样例输入”给出了一组满足“输入格式”要求的输入的例子。

这里给出的输入只是可能用来测试你的程序的一个输入，在测试的时候，还会有更多的输入用来测试你的程序。

样例输出

57

说明：“样例输出”给出了一组满足“输出格式”要求的输出的例子。

样例输出中的结果是和样例输入中的是对应的，因此，你可以使用样例的输入输出简单的检查你的程序。

要特别指出的是，能够通过样例输入输出的程序并不一定是正确的程序，在测试的时候，会用很多组数据进行测试，而不局限于样例数据。有可能一个程序通过了样例数据，但测试的时候仍只能得0分，可能因为这个程序只在一些类似样例的特例中正确，而不具有通用性，再测试更多数据时会出现错误。

比如，对于本题，如果你写一个程序不管输入是什么都输入57，则样例数据是对的，但是测试其他数据，哪怕输入是1和2，这个程序也输出57，则对于其他数据这个程序都不正确。

数据规模与约定

-10000 <= A, B <= 10000。

说明：“数据规模与约定”中给出了试题中主要参数的范围。

这个范围对于解题非常重要，不同的数据范围会导致试题需要使用不同的解法来解决。比如本题中给的A、B范围不大，可以使用整型(int)来保存，如果范围更大，超过int的范围，则要考虑其他方法来保存大数。

有一些范围在方便的时候是在“问题描述”中直接给的，所以在做题时不仅要看这个范围，还要注意问题描述。

提示

本题的C++源代码如下：

1. #include <iostream>
3. **using namespace** std**;**
5. **int** main**()**
6. **{**
7. **int** a**,** b**;**
8. cin **>>** a **>>** b**;**
9. cout **<<** a **+** b**;**
10. **return** 0**;**
11. **}**

本题的C源代码如下：

1. #include <stdio.h>
3. **int** main**()**
4. **{**
5. **int** a**,** b**;**
6. scanf**(**"%d%d", **&**a**,** **&**b**);**
7. printf**(**"%d"**,** a**+**b**);**
8. **return** 0**;**
9. **}**

本题的Java源代码如下：

1. **import** java.util.\*;
3. **public** **class** Main
4. {
5. **public** **static** **void** main(String args[])
6. {
7. Scanner sc = **new** Scanner(System.in);
8. Integer a = sc.nextInt();
9. Integer b = sc.nextInt();
10. System.out.println(a + b);
11. }
12. }

说明：要答题，请点击页面上方的“提交此题”按钮，页面将跳转到提交代码的页面，选择好你的编译语言，将你的编写好的代码粘贴到代码框中，再点击“提交答案”即可。

你的答案提交给系统后系统会自动对你的代码进行判分，并跳转到结果的列表里面，你可以直接从列表中看到你提交的代码的状态，一般几秒钟后就可以看到判分的结果。

本题作为第一题，在提示中已经分别给了C++和Java的代码，你可以直接把这个代码拷贝过去作为自己的代码提交。

请特别注意，Java的主类名必须是Main。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int a, b;**

**cin >> a >> b;**

**cout << a + b;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int a, b;**

**scanf("%d%d", &a, &b);**

**printf("%d", a+b);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**class Main**

**{**

**public static void main(String[] args) throws IOException**

**{**

**args = new BufferedReader(new InputStreamReader(System.in)).readLine().split(" ");**

**System.out.println(Integer.parseInt(args[0])+Integer.parseInt(args[1]));**

**}**

**}**

BASIC-13**基础练习 数列排序**

时间限制：1.0s   内存限制：512.0MB

关键字：数组 排序

问题描述

　　给定一个长度为n的数列，将这个数列按从小到大的顺序排列。1<=n<=200

输入格式

　　第一行为一个整数n。  
　　第二行包含n个整数，为待排序的数，每个整数的绝对值小于10000。

输出格式

　　输出一行，按从小到大的顺序输出排序后的数列。

样例输入

5  
8 3 6 4 9

样例输出

3 4 6 8 9

**参考代码：**

**C++**

**#include<iostream>**

**#include<algorithm>**

**using namespace std;**

**int cmp(int a,int b)**

**{**

**return a<b;**

**}**

**int main()**

**{**

**int n;**

**while(cin>>n)**

**{**

**int a[205];**

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

**{**

**cin>>a[i];**

**}**

**sort(a,a+n,cmp);**

**cout<<a[0];**

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

**{**

**cout<<' '<<a[i];**

**}**

**cout<<endl;**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**void Print(int \*data,int n)**

**{**

**int i;**

**for(i=0;i<n;i++)**

**{**

**printf("%d ",data[i]);**

**}**

**printf("\n");**

**}**

**int Split(int \*data,int pre,int rear)**

**{**

**int value=data[pre];**

**while(pre<rear)**

**{**

**while(data[rear]>=value && pre<rear) rear--;**

**data[pre]=data[rear];**

**while(data[pre]<value && pre<rear) pre++;**

**data[rear]=data[pre];**

**}**

**data[pre]=value;**

**return pre;**

**}**

**//快速排序**

**void QuickSort(int \*data,int pre,int rear)**

**{**

**if(pre<rear)**

**{**

**int mid=Split(data,pre,rear);**

**QuickSort(data,pre,mid-1);**

**QuickSort(data,mid+1,rear);**

**}**

**}**

**int main()**

**{**

**int i;**

**int n;**

**int \*data;**

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

**data=(int \*)malloc(sizeof(int)\*n);**

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&data[i]);**

**}**

**QuickSort(data,0,n-1);**

**Print(data,n);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.util.ArrayList;**

**import java.util.Arrays;**

**public class Main {**

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

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

**int num =Integer.parseInt(bf.readLine());**

**String s = bf.readLine();**

**int arr [] = sort(s);**

**for (int i = 0; i < num; i++) {**

**if(Math.abs(arr[i])>10000){**

**continue;**

**}**

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

**}**

**}**

**private static int [] sort(String s) {**

**String [] str = s.split(" ");**

**int [] arr = new int[str.length];**

**for (int i = 0; i < str.length; i++) {**

**arr[i] = Integer.parseInt(str[i]);**

**}**

**for (int i = 0; i < arr.length - 1; i++) {**

**for (int j = i+1; j < arr.length; j++) {**

**if(arr[i] > arr[j]){**

**int temp = arr[i];**

**arr[i] = arr[j];**

**arr[j] = temp;**

**}**

**}**

**}**

**return arr;**

**}**

**}**

BASIC-12**基础练习 十六进制转八进制**

时间限制：1.0s   内存限制：512.0MB

关键字：进制转换 字符 循环

问题描述

　　给定n个十六进制正整数，输出它们对应的八进制数。

输入格式

　　输入的第一行为一个正整数n （1<=n<=10）。  
　　接下来n行，每行一个由0~9、大写字母A~F组成的字符串，表示要转换的十六进制正整数，每个十六进制数长度不超过100000。

输出格式

　　输出n行，每行为输入对应的八进制正整数。

注意

　　输入的十六进制数不会有前导0，比如012A。  
　　输出的八进制数也不能有前导0。

样例输入

2  
39  
123ABC

样例输出

71  
4435274

提示

　　先将十六进制数转换成某进制数，再由某进制数转换成八进制。

**参考代码：**

**C++**

**#include <iostream>**

**#include <stdio.h>**

**#include <string.h>**

**#include <STDLIB.H>**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int GetI(char c)**

**{**

**return c>>4&1?c&15:(c&15)+9;**

**}**

**int main(int argc, char \*argv[]) {**

**char arr[200001] = {'\0'};**

**char brr[400001] = {'\0'};**

**int n = 0;**

**int i = 0;**

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

**for(i = 0;i < n;i++)**

**{**

**scanf("%s",arr);**

**int m[3] = {1,16,256};**

**int len = strlen(arr);**

**int j = len-1;**

**int a,b,c;**

**a = b = c = 0;**

**int k = 0,l = 0;**

**int count = 0;**

**while(j>-1)**

**{**

**a += (arr[j]>>4&1?arr[j]&15:(arr[j]&15)+9)\*m[k]; //个位**

**if(k==2||j==0)**

**{**

**while(a)**

**{**

**brr[l++] = ((a&7)|48);**

**a = a>>3;**

**count++;**

**}**

**while(j!=0&&count<4)**

**{**

**brr[l++] = '0';**

**count++;**

**}**

**count = 0;**

**}**

**k = (k+1)%3;**

**j--;**

**}**

**strrev(brr);**

**printf("%s\n",brr);**

**memset(arr,'\0',(sizeof(char)\*200001));**

**memset(brr,'\0',(sizeof(char)\*400001));**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <string.h>**

**#define MaxSize 100000**

**void saveB(char \*b,char c3,char c2,char c1,char c0,int start)**

**{**

**b[start]=c3;**

**b[start+1]=c2;**

**b[start+2]=c1;**

**b[start+3]=c0;**

**}**

**int htob(char \*h,char \*b)**

**{**

**int i,j;**

**int hl=strlen(h);**

**for(i=0;i<hl;i++)**

**switch(h[i])**

**{**

**case '0':**

**{**

**saveB(b,'0','0','0','0',4\*i);**

**break;**

**}**

**case '1':**

**{**

**saveB(b,'0','0','0','1',4\*i);**

**break;**

**}**

**case '2':**

**{**

**saveB(b,'0','0','1','0',4\*i);**

**break;**

**}**

**case '3':**

**{**

**saveB(b,'0','0','1','1',4\*i);**

**break;**

**}**

**case '4':**

**{**

**saveB(b,'0','1','0','0',4\*i);**

**break;**

**}**

**case '5':**

**{**

**saveB(b,'0','1','0','1',4\*i);**

**break;**

**}**

**case '6':**

**{**

**saveB(b,'0','1','1','0',4\*i);**

**break;**

**}**

**case '7':**

**{**

**saveB(b,'0','1','1','1',4\*i);**

**break;**

**}**

**case '8':**

**{**

**saveB(b,'1','0','0','0',4\*i);**

**break;**

**}**

**case '9':**

**{**

**saveB(b,'1','0','0','1',4\*i);**

**break;**

**}**

**case 'A':**

**{**

**saveB(b,'1','0','1','0',4\*i);**

**break;**

**}**

**case 'B':**

**{**

**saveB(b,'1','0','1','1',4\*i);**

**break;**

**}**

**case 'C':**

**{**

**saveB(b,'1','1','0','0',4\*i);**

**break;**

**}**

**case 'D':**

**{**

**saveB(b,'1','1','0','1',4\*i);**

**break;**

**}**

**case 'E':**

**{**

**saveB(b,'1','1','1','0',4\*i);**

**break;**

**}**

**case 'F':**

**{**

**saveB(b,'1','1','1','1',4\*i);**

**break;**

**}**

**}**

**return 4\*hl;**

**}**

**int btoo(char \*b,char \*o,int bl)**

**{**

**int i,j;**

**int ol;**

**int value;**

**if(bl%3==0)**

**ol=bl/3;**

**else**

**ol=bl/3+1;**

**j=bl-1;**

**for(i=ol-1;i>=0;i--)**

**{**

**if(i>0)**

**o[i]=b[j]-48+(b[j-1]-48)\*2+(b[j-2]-48)\*4+48;**

**else**

**{**

**switch(j)**

**{**

**case 2:**

**o[i]=b[j]-48+(b[j-1]-48)\*2+(b[j-2]-48)\*4+48;**

**break;**

**case 1:**

**o[i]=b[j]-48+(b[j-1]-48)\*2+48;**

**break;**

**case 0:**

**o[i]=b[j];**

**break;**

**}**

**}**

**j=j-3;**

**}**

**return ol;**

**}**

**void printO(char \*o,int ol)**

**{**

**int i=0;**

**if(o[0]=='0')**

**i=1;**

**for(;i<ol;i++)**

**{**

**printf("%c",o[i]);**

**}**

**printf("\n");**

**}**

**main()**

**{**

**char h[MaxSize];**

**char b[4\*MaxSize];**

**char o[4\*MaxSize/3+1];**

**int n,i,bl,ol;**

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

**getchar();**

**for(i=0;i<n;i++)**

**{**

**gets(h);**

**bl=htob(h,b);**

**ol=btoo(b,o,bl);**

**printO(o,ol);**

**}**

**}**

**JAVA**

**import java.io.BufferedInputStream;**

**import java.io.PrintWriter;**

**import java.math.BigInteger;**

**import java.util.Arrays;**

**import java.util.Comparator;**

**import java.util.PriorityQueue;**

**import java.util.Queue;**

**import java.util.Scanner;**

**public class Main {**

**public void work() {**

**Scanner cin = new Scanner(new BufferedInputStream(System.in));**

**PrintWriter out = new PrintWriter(System.out);**

**int cas;**

**cas = cin.nextInt();**

**int s[] = new int[200000 \* 4 + 10];**

**int ss[] = new int[200000 \* 4 / 3 + 10];**

**int dep, ndep, turn;**

**while (cas -- > 0) {**

**String st = cin.next();**

**dep = 0;**

**for (int i = 0; i < st.length(); ++i) {**

**char c = st.charAt(i);**

**if (c >= 'A' && c <= 'Z')**

**turn = 10 + (c - 'A');**

**else**

**turn = c - '0';**

**for (int j = 3; j >= 0; --j)**

**s[dep ++] = turn >> j & 1;**

**}**

**ndep = 0;**

**for (int i = dep - 1; i >= 0; i -= 3) {**

**turn = 0;**

**for (int j = 0; j < 3 && i - j >= 0; ++j)**

**turn += s[i - j] \* (1 << j);**

**ss[ndep ++] = turn;**

**}**

**while (ndep > 1 && ss[ndep - 1] == 0)**

**-- ndep;**

**for (int i = ndep - 1; i >= 0; -- i)**

**out.print(ss[i]);**

**out.println("");**

**}**

**out.close();**

**}**

**public static void main(String[] args) {**

**Main e = new Main();**

**e.work();**

**}**

**}**

BASIC-11**基础练习 十六进制转十进制**

时间限制：1.0s   内存限制：512.0MB

关键字：进制转换 字符处理 判断

问题描述

　　从键盘输入一个不超过8位的正的十六进制数字符串，将它转换为正的十进制数后输出。  
　　注：十六进制数中的10~15分别用大写的英文字母A、B、C、D、E、F表示。

样例输入

FFFF

样例输出

65535

**参考代码：**

**C++**

**#include<iostream>**

**#include<string>**

**using namespace std;**

**int main()**

**{**

**string s;**

**while(cin>>s)**

**{**

**int leth=s.length();**

**long long sum=0;**

**for(int i=0;i<leth;i++)**

**{**

**if(s[i]>='A'&&s[i]<='F')**

**{**

**sum=sum\*16+s[i]-'A'+10;**

**// cout<<sum<<endl;**

**}**

**else**

**{**

**sum=sum\*16+s[i]-'0';**

**//cout<<sum<<endl;**

**}**

**}**

**cout<<sum<<endl;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**#include<math.h>**

**int main()**

**{**

**double sum=0,x;**

**char a[8];**

**int len,i=0;**

**gets(a);**

**len=strlen(a);**

**while(len)**

**{**

**if(a[len-1]>='A'&&a[len-1]<='F')**

**x=(a[len-1]-'7')\*pow(16,i++);**

**else**

**x=(a[len-1]-'0')\*pow(16,i++);**

**sum+=x;**

**len--;**

**}**

**printf("%.0lf",sum);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main (String args[])throws Exception{**

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**String s=buf.readLine();**

**char c[]=s.toCharArray();**

**int n=c.length;**

**long sum=0;**

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

**if((int)c[i]-48>9){**

**if(c[i]=='A')**

**sum+=10\*Math.pow(16,n-i-1);**

**else if(c[i]=='B')**

**sum+=11\*Math.pow(16,n-i-1);**

**else if(c[i]=='C')**

**sum+=12\*Math.pow(16,n-i-1);**

**else if(c[i]=='D')**

**sum+=13\*Math.pow(16,n-i-1);**

**else if(c[i]=='E')**

**sum+=14\*Math.pow(16,n-i-1);**

**else if(c[i]=='F')**

**sum+=15\*Math.pow(16,n-i-1);**

**}**

**else{**

**sum+=((int)c[i]-48)\*Math.pow(16,n-i-1);**

**}**

**}**

**System.out.println(sum);**

**}**

**}**

BASIC-10**基础练习 十进制转十六进制**

时间限制：1.0s   内存限制：512.0MB

关键字：循环 整除 求余 判断

问题描述

　　十六进制数是在程序设计时经常要使用到的一种整数的表示方式。它有0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F共16个符号，分别表示十进制数的0至15。十六进制的计数方法是满16进1，所以十进制数16在十六进制中是10，而十进制的17在十六进制中是11，以此类推，十进制的30在十六进制中是1E。  
　　给出一个非负整数，将它表示成十六进制的形式。

输入格式

　　输入包含一个非负整数a，表示要转换的数。0<=a<=2147483647

输出格式

　　输出这个整数的16进制表示

样例输入

30

样例输出

1E

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**using namespace std;**

**int main()**

**{**

**int n;**

**char s[100000];**

**while(cin>>n)**

**{**

**int k=0;**

**if(n==0)**

**{**

**cout<<0;**

**}**

**else**

**{**

**while(n!=0)**

**{**

**if(n%16>=10)**

**{**

**s[k++]='A'+n%16-10;**

**}**

**else**

**{**

**s[k++]='0'+n%16;**

**}**

**n=n/16;**

**}**

**for(int i=k-1;i>=0;i--)**

**{**

**cout<<s[i];**

**}**

**}**

**cout<<endl;**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**char data[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};**

**//输出**

**void Print(char \*num,int n)**

**{**

**int i;**

**for(i=n-1;i>=0;i--) printf("%c",num[i]);**

**printf("\0");**

**printf("\n");**

**}**

**//将十六进制数转换为十进制数**

**int Transform(char \*num,long long value)**

**{**

**int n=0;**

**while(value>=16)**

**{**

**num[n++]=data[value%16];**

**value/=16;**

**}**

**num[n++]=data[value%16];**

**return n;**

**}**

**int main()**

**{**

**long long value;**

**char num[10];**

**int n;**

**scanf("%I64d",&value);**

**n=Transform(num,value);**

**Print(num,n);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

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

**BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));**

**printToHexString(buf.readLine());**

**}**

**public static void printToHexString(String s) {**

**StringBuffer st = new StringBuffer();**

**int n = Integer.valueOf(s);**

**if(n == 0) {**

**System.out.println(0);**

**return;**

**}**

**int m;**

**while(n > 0) {**

**m = n % 16;**

**n = n / 16;**

**switch(m) {**

**case 0:**

**st.insert(0, "0");**

**break;**

**case 1:**

**st.insert(0, "1");**

**break;**

**case 2:**

**st.insert(0, "2");**

**break;**

**case 3:**

**st.insert(0, "3");**

**break;**

**case 4:**

**st.insert(0, "4");**

**break;**

**case 5:**

**st.insert(0, "5");**

**break;**

**case 6:**

**st.insert(0, "6");**

**break;**

**case 7:**

**st.insert(0, "7");**

**break;**

**case 8:**

**st.insert(0, "8");**

**break;**

**case 9:**

**st.insert(0, "9");**

**break;**

**case 10:**

**st.insert(0, "A");**

**break;**

**case 11:**

**st.insert(0, "B");**

**break;**

**case 12:**

**st.insert(0, "C");**

**break;**

**case 13:**

**st.insert(0, "D");**

**break;**

**case 14:**

**st.insert(0, "E");**

**break;**

**case 15:**

**st.insert(0, "F");**

**break;**

**}**

**}**

**System.out.println(st);**

**}**

**}**

BASIC-9**基础练习 特殊回文数**

时间限制：1.0s   内存限制：512.0MB

关键字：回文数 循环 条件语句

问题描述

　　123321是一个非常特殊的数，它从左边读和从右边读是一样的。  
　　输入一个正整数n， 编程求所有这样的五位和六位十进制数，满足各位数字之和等于n 。

输入格式

　　输入一行，包含一个正整数n。

输出格式

　　按从小到大的顺序输出满足条件的整数，每个整数占一行。

样例输入

52

样例输出

899998  
989989  
998899

数据规模和约定

　　1<=n<=54。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int n,a,b,c,t;**

**cin>>n;**

**for(a=1;a<10;a++)**

**for(b=0;b<10;b++)**

**for(c=0;c<10;c++)**

**{**

**t=a\*10001+b\*1010+c\*100;**

**if(2\*a+2\*b+c==n)**

**cout<<t<<endl;**

**}**

**for(a=1;a<10;a++)**

**for(b=0;b<10;b++)**

**for(c=0;c<10;c++)**

**{**

**t=a\*100001+b\*10010+c\*1100;**

**if(2\*a+2\*b+2\*c==n)**

**cout<<t<<endl;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int a,b,c,d,e,f,t,all;**

**scanf("%d",&t);**

**for(a=1;a<10;a++)**

**for(b=0;b<10;b++)**

**for(c=0;c<10;c++)**

**for(d=0;d<10;d++)**

**for(e=0;e<10;e++)**

**{**

**if(a==e)**

**if(b==d)**

**{**

**all=a+b+c+d+e;**

**if(all==t)**

**printf("%d\n",a\*10000+b\*1000+c\*100+d\*10+e);**

**}**

**}**

**for(a=1;a<10;a++)**

**for(b=0;b<10;b++)**

**for(c=0;c<10;c++)**

**for(d=0;d<10;d++)**

**for(e=0;e<10;e++)**

**for(f=0;f<10;f++)**

**{**

**if(a==f)**

**if(b==e)**

**if(c==d)**

**{**

**all=a+b+c+d+e+f;**

**if(all==t)**

**printf("%d\n",a\*100000+b\*10000+c\*1000+d\*100+e\*10+f);**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.InputStreamReader;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class Main**

**{**

**public static void main(String[] args) throws Exception**

**{**

**BufferedReader bufferedReader = new BufferedReader(**

**new InputStreamReader(System.in));**

**int n = Integer.parseInt(bufferedReader.readLine());**

**List<String> list\_5 = new ArrayList<String>();**

**List<String> list\_6 = new ArrayList<String>();**

**for (int i = 1; i <= 9; i++)**

**{**

**for (int j = 0; j <= 9; j++)**

**{**

**for (int k = 0; k <= 9; k++)**

**{**

**if (2 \* i + 2 \* j + k == n)**

**{**

**list\_5.add(i + "" + j + "" + k + "" + j + "" + i + "");**

**}**

**if (2 \* i + 2 \* j + 2 \* k == n)**

**{**

**list\_6.add(i + "" + j + "" + k + "" + k + "" + j + ""**

**+ i + "");**

**}**

**}**

**}**

**}**

**list\_5.addAll(list\_6);**

**for (String string : list\_5)**

**{**

**System.out.println(string);**

**}**

**}**

**}**

BASIC-8**基础练习 回文数**

时间限制：1.0s   内存限制：512.0MB

关键字：循环 判断 回文数

问题描述

　　1221是一个非常特殊的数，它从左边读和从右边读是一样的，编程求所有这样的四位十进制数。

输出格式

　　按从小到大的顺序输出满足条件的四位十进制数。

**参考代码：**

**C++**

**#include<stdio.h>**

**int main()**

**{**

**for(int i1=1;i1<10;i1++)**

**{**

**for(int i2=0;i2<10;i2++)**

**{**

**for(int i3=0;i3<10;i3++)**

**{**

**for(int i4=0;i4<10;i4++)**

**{**

**if(i1==i4 && i2==i3)**

**printf("%d%d%d%d\n",i1,i2,i3,i4);**

**}**

**}**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int a,b,c,d,i;**

**for(i=1000;i<=9999;i++)**

**{**

**a=i/1000;**

**b=i/100%10;**

**c=i/10%10;**

**d=i%10;**

**if(a==d&&b==c)**

**printf("%d\n",i);**

**}**

**return 0;**

**}**

**JAVA**

**public class Main**

**{**

**public static void main(String[] args)**

**{**

**for(int i = 1001; i <= 9999; i++)**

**if(isP(i))**

**System.out.println(i);**

**}**

**public static boolean isP(int n)**

**{**

**int s = 0;**

**int p = n;**

**while (n>0)**

**{**

**s = s \* 10 + n % 10;**

**n /= 10;**

**}**

**return s == p;**

**}**

**}**

BASIC-7**基础练习 特殊的数字**

时间限制：1.0s   内存限制：512.0MB

关键字：循环 判断 数位

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int i,j,k;**

**for(i=1;i<=9;i++)**

**{**

**for(j=0;j<=9;j++)**

**{**

**for(k=0;k<=9;k++)**

**{**

**if(i\*100+j\*10+k==i\*i\*i+j\*j\*j+k\*k\*k)**

**{**

**cout<<i\*100+j\*10+k<<endl;**

**}**

**}**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int i,j,k,n;**

**for(n=100;n<1000;n++)**

**{**

**i=n/100;**

**j=n/10%10;**

**k=n%10;**

**if(i\*100+j\*10+k==i\*i\*i+j\*j\*j+k\*k\*k)**

**printf("%d\n",n);**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Arrays;**

**public class Main{**

**public static void main(String args[]){**

**StringBuffer string = new StringBuffer();**

**for(int a=1;a<10;a++){**

**for(int b=0;b<10;b++){**

**for(int c=0;c<10;c++){**

**if(Math.pow(a, 3)+Math.pow(b, 3)+Math.pow(c,3)==(a\*100+b\*10+c)){**

**string.append(a\*100+b\*10+c+",");**

**}**

**}**

**}**

**}**

**String[] str = string.toString().split(",");**

**int[] num = new int[str.length];**

**for(int i=0;i<str.length;i++){**

**num[i] = Integer.parseInt(str[i]);**

**}**

**Arrays.sort(num);**

**for(int i:num){**

**System.out.println(i);**

**}**

**}**

**}**

BASIC-6**基础练习 杨辉三角形**

时间限制：1.0s   内存限制：256.0MB

关键字：基础练习 二维数组

问题描述

杨辉三角形又称Pascal三角形，它的第i+1行是(a+b)i的展开式的系数。

它的一个重要性质是：三角形中的每个数字等于它两肩上的数字相加。

下面给出了杨辉三角形的前4行：

   1

  1 1

 1 2 1

1 3 3 1

给出n，输出它的前n行。

输入格式

输入包含一个数n。

输出格式

输出杨辉三角形的前n行。每一行从这一行的第一个数开始依次输出，中间使用一个空格分隔。请不要在前面输出多余的空格。

样例输入

4

样例输出

1  
1 1  
1 2 1  
1 3 3 1

数据规模与约定

1 <= n <= 34。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**const int MAXN = 40;**

**int n;**

**int a[MAXN][MAXN];**

**int main()**

**{**

**cin >> n;**

**a[0][0] = 1;**

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

**{**

**a[i][0] = a[i][i] = 1;**

**for (int j = 1; j < i; ++j)**

**a[i][j] = a[i-1][j-1] + a[i-1][j];**

**}**

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

**{**

**for (int j = 0; j <= i; ++j)**

**cout << a[i][j] << " ";**

**cout << endl;**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define MAXN 40**

**int n;**

**int a[MAXN][MAXN];**

**int main()**

**{**

**int i, j;**

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

**a[0][0] = 1;**

**for (i = 0; i < n; ++i)**

**{**

**a[i][0] = a[i][i] = 1;**

**for (j = 1; j < i; ++j)**

**a[i][j] = a[i-1][j-1] + a[i-1][j];**

**}**

**for (i = 0; i < n; ++i)**

**{**

**for (j = 0; j <= i; ++j)**

**printf("%d ", a[i][j]);**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.InputStreamReader;**

**import java.io.IOException;**

**public class Main{**

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

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**int n=Integer.parseInt(br.readLine());**

**int[][] a=new int[34][34];**

**int i=0,j=1;**

**while(i<34){**

**a[i][0]=1;**

**i++;**

**}**

**while(j<34){**

**a[j][j]=1;**

**j++;**

**}**

**for(i=2;i<34;i++)**

**for(j=1;j<i;j++)**

**a[i][j]=a[i-1][j-1]+a[i-1][j];**

**for(i=0;i<n;i++)**

**{for(j=0;j<=i;j++)**

**System.out.print(a[i][j]+" ");**

**System.out.println(" ");}**

**}**

**}**

BASIC-5**基础练习 查找整数**

时间限制：1.0s   内存限制：256.0MB

关键字：循环 判断

问题描述

给出一个包含n个整数的数列，问整数a在数列中的第一次出现是第几个。

输入格式

第一行包含一个整数n。

第二行包含n个非负整数，为给定的数列，数列中的每个数都不大于10000。

第三行包含一个整数a，为待查找的数。

输出格式

如果a在数列中出现了，输出它第一次出现的位置(位置从1开始编号)，否则输出-1。

样例输入

6  
1 9 4 8 3 9  
9

样例输出

2

数据规模与约定

1 <= n <= 1000。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**const int MAXN = 10001;**

**int n, a, ans;**

**int s[MAXN];**

**int main()**

**{**

**cin >> n;**

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

**cin >> s[i];**

**cin >> a;**

**ans = -1;**

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

**{**

**if (s[i] == a)**

**{**

**ans = i + 1;**

**break;**

**}**

**}**

**cout << ans << endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define MAXN 10001**

**int n, a, ans;**

**int s[MAXN];**

**int main()**

**{**

**int i;**

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

**for (i = 0; i < n; ++i)**

**scanf("%d", &s[i]);**

**scanf("%d", &a);**

**ans = -1;**

**for (i = 0; i < n; ++i)**

**{**

**if (s[i] == a)**

**{**

**ans = i + 1;**

**break;**

**}**

**}**

**printf("%d", ans);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.Arrays;**

**import java.util.Scanner;**

**import java.io.InputStreamReader;**

**public class Main**

**{**

**static int fun(int[] num,int nn)**

**{**

**for(int i=0;i<num.length;i++)**

**{**

**if(num[i]==nn)**

**return i+1;**

**}**

**return -1;**

**}**

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

**{**

**BufferedReader bf1=new BufferedReader(new InputStreamReader(System.in));**

**int n=Integer.parseInt(bf1.readLine());**

**String[] s1=bf1.readLine().split(" ");**

**int[] num=new int[n];**

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

**num[i]=Integer.parseInt(s1[i]);**

**int nn=Integer.parseInt(bf1.readLine());**

**System.out.println(fun(num, nn));**

**}**

**}**

BASIC-4**基础练习 数列特征**

时间限制：1.0s   内存限制：256.0MB

关键字：循环 最大值 最小值 累加

问题描述

给出n个数，找出这n个数的最大值，最小值，和。

输入格式

第一行为整数n，表示数的个数。

第二行有n个数，为给定的n个数，每个数的绝对值都小于10000。

输出格式

输出三行，每行一个整数。第一行表示这些数中的最大值，第二行表示这些数中的最小值，第三行表示这些数的和。

样例输入

5  
1 3 -2 4 5

样例输出

5  
-2  
11

数据规模与约定

1 <= n <= 10000。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**#include<algorithm>**

**#include<cstring>**

**using namespace std;**

**int main(){**

**int n;**

**while(cin>>n){**

**int a[10005];**

**int sum=0;**

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

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

**sum+=a[i];**

**}**

**sort(a,a+n);**

**cout<<a[n-1]<<endl<<a[0]<<endl<<sum<<endl;**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**#define MAXSIZE 10000**

**int main()**

**{**

**int i;**

**int n;**

**int value;**

**int max=-MAXSIZE;**

**int min=MAXSIZE;**

**int sum=0;**

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

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&value);**

**if(value>max) max=value;**

**if(value<min) min=value;**

**sum+=value;**

**}**

**printf("%d\n",max);**

**printf("%d\n",min);**

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

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main{**

**public static void main(String args[])throws Exception{**

**int n;**

**int sum=0;**

**String str=null;**

**String str2=null;**

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**str=buf.readLine();**

**n=Integer.parseInt(str);**

**str2=buf.readLine();**

**String a[]=str2.split(" ");**

**int array2[]=new int[n];**

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

**array2[i]=Integer.parseInt(a[i]);**

**sum=sum+array2[i];**

**}**

**java.util.Arrays.sort(array2);**

**System.out.println(array2[n-1]);**

**System.out.println(array2[0]);**

**System.out.println(sum);**

**}**

**}**

BASIC-3**基础练习 字母图形**

时间限制：1.0s   内存限制：256.0MB

关键字：循环 字符串

问题描述

利用字母可以组成一些美丽的图形，下面给出了一个例子：

ABCDEFG

BABCDEF

CBABCDE

DCBABCD

EDCBABC

这是一个5行7列的图形，请找出这个图形的规律，并输出一个n行m列的图形。

输入格式

输入一行，包含两个整数n和m，分别表示你要输出的图形的行数的列数。

输出格式

输出n行，每个m个字符，为你的图形。

样例输入

5 7

样例输出

ABCDEFG  
BABCDEF  
CBABCDE  
DCBABCD  
EDCBABC

数据规模与约定

1 <= n, m <= 26。

**参考代码：**

**C++**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int main()**

**{**

**int n, m;**

**cin >> n >> m;**

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

**{**

**for (int j = 0; j < m; ++j)**

**cout << char('A'+abs(i-j));**

**cout << endl;**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**#define MAXSIZE 26**

**//输出**

**void Print(char matrix[MAXSIZE][MAXSIZE],int n,int m)**

**{**

**int i,j;**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**printf("%c",matrix[i][j]);**

**}**

**printf("\n");**

**}**

**}**

**//实现字母图形**

**void Drawing(int n,int m)**

**{**

**int i,j;**

**int point=0;**

**char str;**

**char matrix[MAXSIZE][MAXSIZE];**

**for(i=0;i<n;i++)**

**{**

**str='A';**

**for(j=i;j<m;j++)**

**{**

**matrix[i][j]=str++;**

**}**

**str='A';**

**for(j=i-1;j>=0;j--)**

**{**

**matrix[i][j]=++str;**

**}**

**}**

**Print(matrix,n,m);**

**}**

**int main()**

**{**

**int n,m;**

**scanf("%d%d",&n,&m);**

**Drawing(n,m);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

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

**BufferedReader strin = new BufferedReader(new InputStreamReader(System.in));**

**String s = strin.readLine();**

**String st[] = s.split(" ");**

**char c[] = new char[Integer.parseInt(st[1])];**

**int time = 0;**

**int timemax = Integer.parseInt(st[0]);**

**for(int i=0;i<c.length;i++){**

**c[i] = (char) ('B'+i);**

**}**

**for(int k=0;k<timemax;k++){**

**for(int j =0;j<c.length;j++){**

**if(j<time){**

**c[j]=(char) (c[j]+1);**

**}**

**if(j>=time){**

**c[j]=(char) (c[j]-1);**

**}**

**}**

**time++;**

**System.out.print(c);**

**System.out.println();**

**}**

**}**

**}**

BASIC-2**基础练习 01字串**

时间限制：1.0s   内存限制：256.0MB

关键字：循环

问题描述

对于长度为5位的一个01串，每一位都可能是0或1，一共有32种可能。它们的前几个是：

00000

00001

00010

00011

00100

请按从小到大的顺序输出这32种01串。

输入格式

本试题没有输入。

输出格式

输出32行，按从小到大的顺序每行一个长度为5的01串。

样例输出

00000  
00001  
00010  
00011  
<以下部分省略>

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**for (int i = 0; i <= 1; ++i)**

**for (int j = 0; j <= 1; ++j)**

**for (int k = 0; k <= 1; ++k)**

**for (int l = 0; l <= 1; ++l)**

**for (int m = 0; m <= 1; ++m)**

**cout << i << j << k << l << m << endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int i, j, k, l, m;**

**for (i = 0; i <= 1; ++i)**

**for (j = 0; j <= 1; ++j)**

**for (k = 0; k <= 1; ++k)**

**for (l = 0; l <= 1; ++l)**

**for (m = 0; m <= 1; ++m)**

**printf("%d%d%d%d%d\n", i, j, k, l, m);**

**return 0;**

**}**

**JAVA**

**public class Main{**

**public static void main (String args[]){**

**for(int a1=0;a1<2;a1++){**

**for(int a2=0;a2<2;a2++){**

**for(int a3=0;a3<2;a3++){**

**for(int a4=0;a4<2;a4++){**

**for(int a5=0;a5<2;a5++){**

**StringBuffer s=new StringBuffer();**

**System.out.println(s.append(a1).append(a2).append(a3).append(a4).append(a5));**

**}**

**}**

**}**

**}**

**}**

**}**

**}**

BASIC-1 **基础练习 闰年判断**

时间限制：1.0s   内存限制：256.0MB

关键字：条件判断

问题描述

给定一个年份，判断这一年是不是闰年。

当以下情况之一满足时，这一年是闰年：

1. 年份是4的倍数而不是100的倍数；

2. 年份是400的倍数。

其他的年份都不是闰年。

输入格式

输入包含一个整数y，表示当前的年份。

输出格式

输出一行，如果给定的年份是闰年，则输出yes，否则输出no。

说明：当试题指定你输出一个字符串作为结果（比如本题的yes或者no，你需要严格按照试题中给定的大小写，写错大小写将不得分。

样例输入

2013

样例输出

no

样例输入

2016

样例输出

yes

数据规模与约定

1990 <= y <= 2050。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int y;**

**cin >> y;**

**if (y%4==0 && y%100!=0 || y%400==0)**

**cout << "yes" << endl;**

**else**

**cout << "no" << endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int y;**

**scanf("%d", &y);**

**if (y%4==0 && y%100!=0 || y%400==0)**

**printf("yes");**

**else**

**printf("no");**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.Scanner;**

**import java.io.InputStreamReader;**

**import java.\*;**

**public class Main**

**{**

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

**{**

**BufferedReader bf1=new BufferedReader(new InputStreamReader(System.in));**

**int num=Integer.parseInt(bf1.readLine());**

**if((num%4==0&&num%100!=0)||(num%400==0))**

**System.out.println("yes");**

**else**

**System.out.println("no");**

**}**

**}**

BASIC-30**基础练习 阶乘计算**

时间限制：1.0s   内存限制：512.0MB

关键字：高精度

问题描述

　　输入一个正整数n，输出n!的值。  
　　其中n!=1\*2\*3\*…\*n。

算法描述

　　n!可能很大，而计算机能表示的整数范围有限，需要使用高精度计算的方法。使用一个数组A来表示一个大整数a，A[0]表示a的个位，A[1]表示a的十位，依次类推。  
　　将a乘以一个整数k变为将数组A的每一个元素都乘以k，请注意处理相应的进位。  
　　首先将a设为1，然后乘2，乘3，当乘到n时，即得到了n!的值。

输入格式

　　输入包含一个正整数n，n<=1000。

输出格式

　　输出n!的准确值。

样例输入

10

样例输出

3628800

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX 10000**

**#define mod 10000**

**#define baselen 4**

**#define in(a) scanf("%d",&a)**

**#define out1(a) printf("%d",a)**

**#define out2(a) printf("%04d",a)**

**typedef int type;**

**struct bint{**

**type dig[MAX], len;**

**bint(){len = 0, dig[0] = 0;}**

**};**

**void by(bint a, type b, bint& c){**

**type i, carry;**

**for( i = carry = 0; i <= a.len || carry; i++){**

**if( i <= a.len ) carry += b\*a.dig[i];**

**c.dig[i] = carry%mod;**

**carry /= mod;**

**}**

**i--;**

**while( i && !c.dig[i] )i--;**

**c.len = i;**

**}**

**bool input(bint& a){**

**type i, j, w, k, p;**

**char data[MAX\*baselen+1];**

**if(scanf("%s",data)==EOF)return false;**

**w = strlen(data) - 1, a.len = 0;**

**for(p=0;p<=w&&data[p]=='0';p++);**

**while(1){**

**i = j = 0, k = 1;**

**while(i<baselen&&w>=p){**

**j = j+ (data[w--] - '0')\*k;**

**k \*= 10, i++;**

**}**

**a.dig[a.len++] = j;**

**if(w<p)break;**

**}**

**a.len--;**

**return true;**

**}**

**void output(bint& a){**

**type i;**

**i = a.len - 1;**

**out1(a.dig[a.len]);**

**while(i>=0)out2(a.dig[i--]);**

**}**

**void give(type a, bint& b){**

**b.dig[0] = a%mod;**

**a /= mod;**

**if(a>0)b.dig[1] = a, b.len = 1;**

**else b.len = 0;**

**}**

**int main()**

**{**

**bint a;int b,i;scanf("%d",&b);give(1,a);**

**for(i=2;i<=b;i++)by(a,i,a);**

**output(a);printf("\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define N 10000**

**int main()**

**{**

**int a[N]={1};**

**int k=0,l=1,n;**

**int i,j;**

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

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

**{**

**for(j=0;j<l;j++)**

**{**

**a[j]=a[j]\*i+k;**

**k=a[j]/10000;**

**a[j]=a[j]%10000;**

**}**

**if(k)**

**{**

**a[j]=k;**

**l++;**

**k=0;**

**}**

**}**

**printf("%d",a[l-1]);**

**for(i=l-2;i>=0;i--)**

**printf("%04d",a[i]);**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.math.BigInteger;**

**import java.util.\*;**

**public class Main{**

**protected static ArrayList table = new ArrayList();**

**static {**

**table.add(BigInteger.valueOf(1));**

**}**

**public static synchronized BigInteger factorial(int x) {**

**for (int size = table.size(); size <= x; size++) {**

**BigInteger lastfact = (BigInteger) table.get(size - 1);**

**BigInteger nextfact = lastfact.multiply(BigInteger.valueOf(size));**

**table.add(nextfact);**

**}**

**return (BigInteger) table.get(x);**

**}**

**public static void main(String[] args) {**

**Scanner s=new Scanner(System.in);**

**int n=s.nextInt();**

**System.out.print(factorial(n));**

**}**

**}**

BASIC-29**基础练习 高精度加法**

时间限制：1.0s   内存限制：512.0MB

关键字：数组 高精度

问题描述

　　输入两个整数a和b，输出这两个整数的和。a和b都不超过100位。

算法描述

　　由于a和b都比较大，所以不能直接使用语言中的标准数据类型来存储。对于这种问题，一般使用数组来处理。  
　　定义一个数组A，A[0]用于存储a的个位，A[1]用于存储a的十位，依此类推。同样可以用一个数组B来存储b。  
　　计算c = a + b的时候，首先将A[0]与B[0]相加，如果有进位产生，则把进位（即和的十位数）存入r，把和的个位数存入C[0]，即C[0]等于(A[0]+B[0])%10。然后计算A[1]与B[1]相加，这时还应将低位进上来的值r也加起来，即C[1]应该是A[1]、B[1]和r三个数的和．如果又有进位产生，则仍可将新的进位存入到r中，和的个位存到C[1]中。依此类推，即可求出C的所有位。  
　　最后将C输出即可。

输入格式

　　输入包括两行，第一行为一个非负整数a，第二行为一个非负整数b。两个整数都不超过100位，两数的最高位都不是0。

输出格式

　　输出一行，表示a + b的值。

样例输入

20100122201001221234567890  
2010012220100122

样例输出

20100122203011233454668012

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**#include<cstring>**

**using namespace std;**

**int a[401],alen,b[401],blen,c[400],clen;**

**char st[400];**

**int main()**

**{**

**int i,j,n,len;**

**scanf("%s",st);**

**alen= strlen(st);**

**for(i=1;i<=alen;i++)**

**a[i]= st[alen-i]-48;**

**scanf("%s",st);**

**blen= strlen(st);**

**for(i=1;i<=blen;i++)**

**b[i]= st[blen-i]-48;**

**if(alen>blen) clen=alen;**

**else clen=blen;**

**for(i=1;i<=clen;i++)**

**c[i]=a[i]+b[i];**

**for(i=1;i<=clen;i++)**

**{**

**if(c[i]>=10)**

**{**

**c[i+1]=c[i+1]+ c[i]/10;**

**c[i]=c[i]%10;**

**}**

**}**

**if(c[clen+1]>0)clen++;**

**for(i=clen;i>=1;i--)**

**{**

**printf("%d",c[i]);**

**}**

**printf("\n");**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**#include<stdlib.h>**

**void f(char a[],char b[])**

**{**

**int w=200,i,j,la,lb;**

**la=strlen(a);lb=strlen(b);**

**char c[200];**

**for(i=la;i<w;i++)a[i]=48;**

**for(i=lb;i<w;i++)b[i]=48;**

**a[w-1]=0;b[w-1]=0;**

**for(i=0;i<la;i++)c[i]=a[i];**

**for(i=0;i<w-1-la;i++)a[i]=48;j=0;**

**for(i=w-1-la;i<w-1;i++){a[i]=c[j];j++;}**

**for(i=0;i<lb;i++)c[i]=b[i];**

**for(i=0;i<w-1-lb;i++)b[i]=48;j=0;**

**for(i=w-1-lb;i<w-1;i++){b[i]=c[j];j++;}**

**for(i=w;i>=0;i--)**

**{**

**j=a[i]+b[i]-96;**

**if(j>9)a[i-1]=a[i-1]+j/10;**

**c[i]=j%10+48;**

**}**

**c[w-1]=0;**

**for(i=0;i<w;i++)if(c[i]!='0')break;**

**for(;i<w-1;i++)printf("%c",c[i]);printf("\n");**

**}**

**int main()**

**{**

**char a[200],b[200];gets(a);gets(b);**

**f(a,b);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String s1 = br.readLine();**

**String s2 = br.readLine();**

**int a[] = new int[100000];**

**int b[] = new int[100000];**

**int l1 = s1.length();**

**int l2 = s2.length();**

**for(int i=0; i<l1; i++){**

**a[l1-1-i] = Character.getNumericValue(s1.charAt(i));**

**}**

**for(int i=0; i<l2; i++){**

**b[l2-1-i] = Character.getNumericValue(s2.charAt(i));**

**}**

**int max = l1>l2?l1:l2;**

**int c[] = new int[100000];**

**for(int i=0; i<max; i++){**

**c[i] = a[i] + b[i];**

**}**

**for(int i=0; i<max-1; i++){**

**if(c[i]>9){**

**c[i+1] += c[i]/10;**

**c[i] = c[i]%10;**

**}**

**}**

**while(c[max-1]>9){**

**c[max] = c[max-1]/10;**

**c[max-1] = c[max-1]%10;**

**max++;**

**}**

**for(int i=max-1; i>=0; i--){**

**System.out.print(c[i]);**

**}**

**}**

**}**

BASIC-28**基础练习 Huffuman树**

时间限制：1.0s   内存限制：512.0MB

关键字：贪心 Huffuman

问题描述

　　Huffman树在编码中有着广泛的应用。在这里，我们只关心Huffman树的构造过程。  
　　给出一列数{pi}={p0, p1, …, pn-1}，用这列数构造Huffman树的过程如下：  
　　1. 找到{pi}中最小的两个数，设为pa和pb，将pa和pb从{pi}中删除掉，然后将它们的和加入到{pi}中。这个过程的费用记为pa + pb。  
　　2. 重复步骤1，直到{pi}中只剩下一个数。  
　　在上面的操作过程中，把所有的费用相加，就得到了构造Huffman树的总费用。  
　　本题任务：对于给定的一个数列，现在请你求出用该数列构造Huffman树的总费用。  
  
　　例如，对于数列{pi}={5, 3, 8, 2, 9}，Huffman树的构造过程如下：  
　　1. 找到{5, 3, 8, 2, 9}中最小的两个数，分别是2和3，从{pi}中删除它们并将和5加入，得到{5, 8, 9, 5}，费用为5。  
　　2. 找到{5, 8, 9, 5}中最小的两个数，分别是5和5，从{pi}中删除它们并将和10加入，得到{8, 9, 10}，费用为10。  
　　3. 找到{8, 9, 10}中最小的两个数，分别是8和9，从{pi}中删除它们并将和17加入，得到{10, 17}，费用为17。  
　　4. 找到{10, 17}中最小的两个数，分别是10和17，从{pi}中删除它们并将和27加入，得到{27}，费用为27。  
　　5. 现在，数列中只剩下一个数27，构造过程结束，总费用为5+10+17+27=59。

输入格式

　　输入的第一行包含一个正整数n（n<=100）。  
　　接下来是n个正整数，表示p0, p1, …, pn-1，每个数不超过1000。

输出格式

　　输出用这些数构造Huffman树的总费用。

样例输入

5  
5 3 8 2 9

样例输出

59

**参考代码：**

**C++**

**#include<iostream>**

**#include<queue>**

**using namespace std;**

**priority\_queue<int, vector<int>, greater<int> > pq; //构造从小到大的优先队列**

**int main() {**

**int n;**

**cin >> n;**

**while (!pq.empty())**

**pq.pop();**

**int x, s;**

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

**cin >> x;**

**pq.push(x);**

**}**

**int sum = 0;**

**while (pq.size() > 1) {**

**s = pq.top();**

**pq.pop();**

**s += pq.top();**

**pq.pop();**

**sum += s;**

**pq.push(s);**

**}**

**cout << sum << endl;**

**}**

**C**

**#include <stdio.h>**

**typedef struct**

**{**

**int a[100];**

**int len;**

**}huf;**

**int sum=0;**

**int del(huf\* in,int t)**

**{**

**int i,j;**

**for(i=0;i<in->len && in->a[i]!=t;i++);**

**for(;i<in->len-1;i++)**

**in->a[i]=in->a[i+1];**

**in->len--;**

**return 1;**

**}**

**int add(huf\* in,int t)**

**{**

**in->a[in->len]=t;**

**in->len++;**

**}**

**int find\_two\_mins(huf\* in)**

**{**

**int i,j,t;**

**int mina,minb;**

**for(i=0;i<in->len-1;i++)**

**for(j=i+1;j<in->len;j++)**

**if(in->a[i]>in->a[j])**

**{**

**t=in->a[i];**

**in->a[i]=in->a[j];**

**in->a[j]=t;**

**}**

**mina=in->a[0];**

**minb=in->a[1];**

**del(in,mina);**

**del(in,minb);**

**add(in,mina+minb);**

**return mina+minb;**

**}**

**int main()**

**{**

**huf in;**

**int i,j,n;**

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

**in.len=n;**

**for(i=0;i<n;i++)**

**scanf("%d",&in.a[i]);**

**while(1)**

**{**

**if(in.len==2)**

**{**

**sum=sum+in.a[0]+in.a[1];**

**break;**

**}**

**sum+=find\_two\_mins(&in);**

**}**

**printf("%d",sum);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.math.BigInteger;**

**public class Main {**

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

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**int n=Integer.parseInt(buf.readLine());**

**String str=buf.readLine();**

**String s[]=str.split(" ");**

**int sum=0;**

**int a[] =new int[n];**

**int num=0;**

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

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

**java.util.Arrays.sort(a);**

**if(n>1){**

**sum=a[0]+a[1];**

**num=sum;**

**a[1]=num;**

**for(int j=2;j<n;j++){**

**if(j<n){**

**java.util.Arrays.sort(a);**

**num=a[j-1]+a[j];**

**a[j]=num;**

**java.util.Arrays.sort(a);**

**sum+=num;**

**}**

**}**

**}**

**else {**

**sum=a[0];**

**}**

**System.out.println(sum);**

**}**

**}**

BASIC-27**基础练习 2n皇后问题**

时间限制：1.0s   内存限制：512.0MB

关键字：八皇后问题 搜索

问题描述

　　给定一个n\*n的棋盘，棋盘中有一些位置不能放皇后。现在要向棋盘中放入n个黑皇后和n个白皇后，使任意的两个黑皇后都不在同一行、同一列或同一条对角线上，任意的两个白皇后都不在同一行、同一列或同一条对角线上。问总共有多少种放法？n小于等于8。

输入格式

　　输入的第一行为一个整数n，表示棋盘的大小。  
　　接下来n行，每行n个0或1的整数，如果一个整数为1，表示对应的位置可以放皇后，如果一个整数为0，表示对应的位置不可以放皇后。

输出格式

　　输出一个整数，表示总共有多少种放法。

样例输入

4  
1 1 1 1  
1 1 1 1  
1 1 1 1  
1 1 1 1

样例输出

2

样例输入

4  
1 0 1 1  
1 1 1 1  
1 1 1 1  
1 1 1 1

样例输出

0

**参考代码：**

**C++**

**#include<cstdio>**

**using namespace std;**

**int n;**

**int sum;**

**bool g[9][9];**

**bool wh[9];**

**bool wd[17];**

**bool wu[17];**

**bool bh[9];**

**bool bd[17];**

**bool bu[17];**

**void white(int h){**

**if(h==n){**

**sum++;**

**}else{**

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

**if(!g[h][i])continue;**

**if(wh[i])continue;**

**if(wd[i+h])continue;**

**if(wu[(i-h)+n])continue;**

**wh[i]=wd[i+h]=wu[(i-h)+n]=1;**

**white(h+1);**

**wh[i]=wd[i+h]=wu[(i-h)+n]=0;**

**}**

**}**

**}**

**void black(int h){**

**if(h==n){**

**white(0);**

**}else{**

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

**if(!g[h][i])continue;**

**if(bh[i])continue;**

**if(bd[i+h])continue;**

**if(bu[(i-h)+n])continue;**

**g[h][i]=0;**

**bh[i]=bd[i+h]=bu[(i-h)+n]=1;**

**black(h+1);**

**g[h][i]=1;**

**bh[i]=bd[i+h]=bu[(i-h)+n]=0;**

**}**

**}**

**}**

**int main(){**

**int i;**

**int x;**

**sum=0;**

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

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

**wh[i]=bh[i]=0;**

**wd[i]=bd[i]=0;**

**wu[i]=bu[i]=0;**

**for(int j=0;j<n;j++){**

**scanf("%d",&x);**

**g[i][j]=(bool)x;**

**}**

**}**

**for(;i<2\*n;i++){**

**wd[i]=bd[i]=0;**

**wu[i]=bu[i]=0;**

**}**

**black(0);**

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

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int sum=0;**

**int p(int n,int x,int y,int m[][8],int s)**

**{**

**int i,j;**

**for(i=x-1;i>=0;i--)**

**{**

**if(m[i][y]==s)return 0;**

**}**

**for(i=x-1,j=y-1;i>=0&&j>=0;i--,j--)**

**{**

**if(m[i][j]==s)return 0;**

**}**

**for(i=x-1,j=y+1;i>=0&&j<n;i--,j++)**

**{**

**if(m[i][j]==s)return 0;**

**}**

**return 1;**

**}**

**int h(int n,int x,int m[][8],int s)**

**{**

**int i;**

**if(x==n)**

**{**

**if(s==2)h(n,0,m,3);**

**else sum++;**

**return 0;**

**}**

**for(i=0;i<n;i++)**

**{**

**if(m[x][i]!=1)continue;**

**if(p(n,x,i,m,s))m[x][i]=s;**

**else continue;**

**h(n,x+1,m,s);**

**m[x][i]=1;**

**}**

**return 0;**

**}**

**int main()**

**{**

**int n,m[8][8],i,j;**

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

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<n;j++)**

**{**

**scanf("%d",&m[i][j]);**

**}**

**}**

**h(n,0,m,2);**

**printf("%d",sum);**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**//2n皇后问题**

**public class Main {**

**public static final int no=0;**

**public static final int yes=1;**

**public static final int hei=2;**

**public static final int bai=3;**

**public static int n;**

**public static int[][] qipan=null;**

**public static int count=0;**

**public static int num=0;**

**public static int get(int qizi){**

**int num=qizi==2? 3: 2;**

**return num;**

**}**

**public static boolean jc(int k,int m,int qizi){**

**if(qipan[k][m]==get(qizi)||qipan[k][m]==no){**

**return false;**

**}**

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

**if(qipan[i][m]==qizi){**

**return false;**

**}**

**}**

**int i=0,j=0;**

**for(i=k,j=m;i<n&&j<n;i++,j++){**

**if(qipan[i][j]==qizi){**

**return false;**

**}**

**}**

**for(i=k,j=m;i>=0&&j>=0;i--,j--){**

**if(qipan[i][j]==qizi){**

**return false;**

**}**

**}**

**for(i=k,j=m;i>=0&&j<n;i--,j++){**

**if(qipan[i][j]==qizi){**

**return false;**

**}**

**}**

**for(i=k,j=m;i<n&&j>=0;i++,j--){**

**if(qipan[i][j]==qizi){**

**return false;**

**}**

**}**

**return true;**

**}**

**public static boolean sum(){**

**int bainum=0;**

**int heinum=0;**

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

**for(int j=0;j<n;j++){**

**if(qipan[i][j]==bai)**

**bainum++;**

**if(qipan[i][j]==hei)**

**heinum++;**

**}**

**}**

**if(bainum==n&&heinum==n){**

**num++;**

**return true;**

**}**

**return false;**

**}**

**public static void mk(int line , int qizi){**

**if(line>=n) return ;**

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

**if(jc(line,i,qizi)){**

**qipan[line][i]=qizi;**

**count++;**

**if(count>=n){**

**if(sum()){**

**}**

**int c=count;**

**count=0;**

**mk(0,bai);**

**count=c;**

**}**

**mk(line+1,qizi);**

**qipan[line][i]=yes;**

**count--;**

**}**

**}**

**}**

**public static void print(){**

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

**for(int j=0;j<n;j++){**

**System.out.print(qipan[i][j]+" ");**

**}**

**System.out.println();**

**}**

**System.out.println(":"+num);**

**}**

**public static void main(String args[]){**

**Scanner scan=new Scanner(System.in);**

**n=scan.nextInt();**

**qipan=new int[n][n];**

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

**for(int j=0;j<n;j++){**

**qipan[i][j]=scan.nextInt();**

**}**

**}**

**mk(0,hei);**

**System.out.println(num);**

**}**

**}**

BASIC-26**基础练习 报时助手**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 条件判断

问题描述

　　给定当前的时间，请用英文的读法将它读出来。  
　　时间用时h和分m表示，在英文的读法中，读一个时间的方法是：  
　　如果m为0，则将时读出来，然后加上“o'clock”，如3:00读作“three o'clock”。  
　　如果m不为0，则将时读出来，然后将分读出来，如5:30读作“five thirty”。  
　　时和分的读法使用的是英文数字的读法，其中0~20读作：  
　　0:zero, 1: one, 2:two, 3:three, 4:four, 5:five, 6:six, 7:seven, 8:eight, 9:nine, 10:ten, 11:eleven, 12:twelve, 13:thirteen, 14:fourteen, 15:fifteen, 16:sixteen, 17:seventeen, 18:eighteen, 19:nineteen, 20:twenty。  
　　30读作thirty，40读作forty，50读作fifty。  
　　对于大于20小于60的数字，首先读整十的数，然后再加上个位数。如31首先读30再加1的读法，读作“thirty one”。  
　　按上面的规则21:54读作“twenty one fifty four”，9:07读作“nine seven”，0:15读作“zero fifteen”。

输入格式

　　输入包含两个非负整数h和m，表示时间的时和分。非零的数字前没有前导0。h小于24，m小于60。

输出格式

　　输出时间时刻的英文。

样例输入

0 15

样例输出

zero fifteen

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**#include <map>**

**using namespace std;**

**int main(int argc, char\*\* argv)**

**{**

**map<int,string> maptime;**

**maptime[0]="zero";**

**maptime[1]="one";**

**maptime[2]="two";**

**maptime[3]="three";**

**maptime[4]="four";**

**maptime[5]="five";**

**maptime[6]="six";**

**maptime[7]="seven";**

**maptime[8]="eight";**

**maptime[9]="nine";**

**maptime[10]="ten";**

**maptime[11]="eleven";**

**maptime[12]="twelve";**

**maptime[13]="thirteen";**

**maptime[14]="fourteen";**

**maptime[15]="fifteen";**

**maptime[16]="sixteen";**

**maptime[17]="seventeen";**

**maptime[18]="eighteen";**

**maptime[19]="nineteen";**

**maptime[20]="twenty";**

**maptime[30]="thirty";**

**maptime[40]="forty";**

**maptime[50]="fifty";**

**int h,m;**

**cin>>h>>m;**

**if(m==0)**

**{**

**if(h<=20)**

**{**

**cout<<maptime[h]<<" o'clock";**

**}**

**else**

**{**

**cout<<maptime[20]<<" "<<maptime[h-20]<<" o'clock";**

**}**

**}**

**else**

**{**

**if(h<=20)**

**{**

**cout<<maptime[h]<<" ";**

**}**

**else**

**{**

**cout<<maptime[20]<<" "<<maptime[h-20]<<" ";**

**}**

**if(m<=20)**

**{**

**cout<<maptime[m]<<" ";**

**}**

**else**

**{**

**int k=m%10;**

**cout<<maptime[m-k]<<" "<<maptime[k]<<" ";**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**int h,m;**

**char g[50][50]={"zero","one","two","three","four","five","six","seven","eight","nine","ten","eleven","twelve","thirteen","fourteen","fifteen","sixteen","seventeen","eighteen","nineteen","twenty"};**

**char s[10][10]={"twenty","thirty","forty","fifty"};**

**scanf("%d%d",&h,&m);**

**if(m==0)if(h<21)printf("%s o'clock",g[h]);else {printf("%s %s o'clock",s[h/10-2],g[h%10]);}**

**else {**

**if(h<21)printf("%s ",g[h]);else {printf("%s %s ",s[h/10-2],g[h%10]);}**

**if(m<21)printf("%s ",g[m]);else {printf("%s %s ",s[m/10-2],g[m%10]);}**

**}**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**String str=buf.readLine();**

**String s[]=str.split(" ");**

**StringBuffer bf=new StringBuffer();**

**int i=Integer.parseInt(s[0]);**

**int n=Integer.parseInt(s[1]);**

**switch(i){**

**case 0:{**

**bf.append("zero ");**

**break;**

**}**

**case 1:{**

**bf.append("one ");**

**break;**

**}**

**case 2:{**

**bf.append("two ");**

**break;**

**}**

**case 3:{**

**bf.append("three ");**

**break;**

**}**

**case 4:{**

**bf.append("four ");**

**break;**

**}**

**case 5:{**

**bf.append("five ");**

**break;**

**}**

**case 6:{**

**bf.append("six ");**

**break;**

**}**

**case 7:{**

**bf.append("seven ");**

**break;**

**}**

**case 8:{**

**bf.append("eight ");**

**break;**

**}**

**case 9:{**

**bf.append("nine ");**

**break;**

**}**

**case 10:{**

**bf.append("ten ");**

**break;**

**}**

**case 11:{**

**bf.append("eleven ");**

**break;**

**}**

**case 12:{**

**bf.append("twelve ");**

**break;**

**}**

**case 13:{**

**bf.append("thirteen ");**

**break;**

**}**

**case 14:{**

**bf.append("fourteen ");**

**break;**

**}**

**case 15:{**

**bf.append("fifteen ");**

**break;**

**}**

**case 16:{**

**bf.append("sixteen ");**

**break;**

**}**

**case 17:{**

**bf.append("seventeen ");**

**break;**

**}**

**case 18:{**

**bf.append("eighteen ");**

**break;**

**}**

**case 19:{**

**bf.append("nineteen ");**

**break;**

**}**

**case 20:{**

**bf.append("twenty ");**

**break;**

**}**

**case 21:{**

**bf.append("twenty one ");**

**break;**

**}**

**case 22:{**

**bf.append("twenty two ");**

**break;**

**}**

**case 23:{**

**bf.append("twenty three ");**

**break;**

**}**

**}**

**if(n==0){**

**bf.append("o'clock");**

**}**

**else if(n/10<2){**

**switch(n){**

**case 1:{**

**bf.append("one");**

**break;**

**}**

**case 2:{**

**bf.append("two");**

**break;**

**}**

**case 3:{**

**bf.append("three");**

**break;**

**}**

**case 4:{**

**bf.append("Zero");**

**break;**

**}**

**case 5:{**

**bf.append("five");**

**break;**

**}**

**case 6:{**

**bf.append("six");**

**break;**

**}**

**case 7:{**

**bf.append("seven");**

**break;**

**}**

**case 8:{**

**bf.append("eight");**

**break;**

**}**

**case 9:{**

**bf.append("nine");**

**break;**

**}**

**case 10:{**

**bf.append("ten");**

**break;**

**}**

**case 11:{**

**bf.append("eleven");**

**break;**

**}**

**case 12:{**

**bf.append("twelve ");**

**break;**

**}**

**case 13:{**

**bf.append("thirteen ");**

**break;**

**}**

**case 14:{**

**bf.append("fourteen ");**

**break;**

**}**

**case 15:{**

**bf.append("fifteen ");**

**break;**

**}**

**case 16:{**

**bf.append("sixteen ");**

**break;**

**}**

**case 17:{**

**bf.append("seventeen ");**

**break;**

**}**

**case 18:{**

**bf.append("eighteen ");**

**break;**

**}**

**case 19:{**

**bf.append("nineteen ");**

**break;**

**}**

**}**

**}**

**else {**

**switch(n/10){**

**case 2:{**

**bf.append("twenty ");**

**break;**

**}**

**case 3:{**

**bf.append("thirty ");**

**break;**

**}**

**case 4:{**

**bf.append("forty ");**

**break;**

**}**

**case 5:{**

**bf.append("fifty ");**

**break;**

**}**

**}**

**switch(n%10){**

**case 1:{**

**bf.append("one");**

**break;**

**}**

**case 2:{**

**bf.append("two");**

**break;**

**}**

**case 3:{**

**bf.append("three");**

**break;**

**}**

**case 4:{**

**bf.append("four");**

**break;**

**}**

**case 5:{**

**bf.append("five");**

**break;**

**}**

**case 6:{**

**bf.append("six");**

**break;**

**}**

**case 7:{**

**bf.append("seven");**

**break;**

**}**

**case 8:{**

**bf.append("eight");**

**break;**

**}**

**case 9:{**

**bf.append("nine");**

**break;**

**}**

**}**

**}**

**System.out.println(bf.toString());**

**}**

**}**

BASIC-25**基础练习 回形取数**

时间限制：1.0s   内存限制：512.0MB

关键字：二维数组 循环

问题描述

　　回形取数就是沿矩阵的边取数，若当前方向上无数可取或已经取过，则左转90度。一开始位于矩阵左上角，方向向下。

输入格式

　　输入第一行是两个不超过200的正整数m, n，表示矩阵的行和列。接下来m行每行n个整数，表示这个矩阵。

输出格式

　　输出只有一行，共mn个数，为输入矩阵回形取数得到的结果。数之间用一个空格分隔，行末不要有多余的空格。

样例输入

3 3  
1 2 3  
4 5 6  
7 8 9

样例输出

1 4 7 8 9 6 3 2 5

样例输入

3 2  
1 2  
3 4  
5 6

样例输出

1 3 5 6 4 2

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstring>**

**#include<cstdlib>**

**#include<iostream>**

**using namespace std;**

**int m, n;**

**int a[210][210]={0};**

**int ans[44100];**

**int t=0;**

**void sr()**

**{**

**scanf("%d %d",&m,&n);**

**int i,j;**

**for(i=0;i<m;i++)for(j=0;j<n;j++)scanf("%d",&a[i][j]);**

**}**

**void xx(int x1,int y1,int x,int y)**

**{**

**int i,j;**

**if(x==0||y==0)return;**

**if(x==1)**

**{**

**for(i=0;i<y;i++)**

**{**

**ans[t]=a[x1][y1];**

**y1++;**

**t++;**

**}**

**return;**

**}**

**if(y==1)**

**{**

**for(i=0;i<x;i++)**

**{**

**ans[t]=a[x1][y1];**

**x1++;**

**t++;**

**}**

**return;**

**}**

**int v=x1,h=y1;**

**ans[t]=a[x1][y1];**

**t++;**

**for(i=0;i<x-1;i++){v++;ans[t]=a[v][h];t++;}**

**for(i=0;i<y-1;i++){h++;ans[t]=a[v][h];t++;}**

**for(i=0;i<x-1;i++){v--;ans[t]=a[v][h];t++;}**

**for(i=0;i<y-2;i++){h--;ans[t]=a[v][h];t++;}**

**xx(v+1,h,x-2,y-2);**

**}**

**void sc()**

**{**

**int c=n\*m;**

**if(c==0)return;**

**printf("%d",ans[0]);**

**for(int i=1;i<c;i++)printf(" %d",ans[i]);**

**}**

**int main()**

**{**

**sr();**

**int x=m,y=n;**

**xx(0,0,x,y);**

**sc();**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**#define MAX\_N 200**

**int m,n;**

**int a[MAX\_N][MAX\_N],b[MAX\_N][MAX\_N];**

**int s=0;**

**void solve(int i,int j)**

**{**

**if(i<m && i>=0 && j<n && j>=0 && b[i][j] == 0)**

**{**

**printf("%d ",a[i][j]);**

**b[i][j] = 1;**

**}**

**else**

**{**

**s++;**

**return ;**

**}**

**if(s%4 == 0)**

**solve(i+1,j);**

**if(s%4 == 1)**

**solve(i,j+1);**

**if(s%4 == 2)**

**solve(i-1,j);**

**if(s%4 == 3)**

**solve(i,j-1);**

**if(s%4 == 0)**

**solve(i+1,j);**

**if(s%4 == 1)**

**solve(i,j+1);**

**if(s%4 == 2)**

**solve(i-1,j);**

**if(s%4 == 3)**

**solve(i,j-1);**

**return ;**

**}**

**int main()**

**{**

**memset(b,0,sizeof(b));**

**scanf("%d%d",&m,&n);**

**int i,j;**

**for(i=0; i<m; i++)**

**for(j=0; j<n; j++)**

**scanf("%d",&a[i][j]);**

**solve(0,0);**

**puts("");**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**String s=buf.readLine();**

**String str[]=s.split(" ");**

**int m=Integer.parseInt(str[0]);**

**int n=Integer.parseInt(str[1]);**

**int a[][]=new int[m][n];**

**int num=m\*n;**

**int i=0,j=0,x=0,y=0,z=0 ,t=1;**

**int p=0;**

**for( int g=0;g<m;g++){**

**String s1=buf.readLine();**

**String str1[]=s1.split(" ");**

**for(int f=0;f<n;f++){**

**int c=Integer.parseInt(str1[f]);**

**a[g][f]=c;**

**}**

**}**

**while(num>0){**

**if(j<n&&x<m){**

**p=a[i][j];**

**if(i<m-1){**

**i++;**

**}**

**x++;**

**}**

**else if(j<n-1&&i==m-1){**

**j++;**

**y++;**

**p=a[i][j];**

**//System.out.println(a[i][j]);**

**}**

**else if(j==y&&i>z){**

**i--;**

**p=a[i][j];**

**// System.out.print(a[i][j]+" ");**

**}**

**else if(i==z&&j>z){**

**j--;**

**p=a[i][j];**

**// System.out.print(a[i][j]+" .");**

**if(i==z&&j==z+1){**

**z++;**

**i++;**

**m=m-1;**

**n=n-1;**

**x=z;**

**y=z;**

**}**

**}**

**num--;**

**System.out.print(p+" ");**

**}**

**}**

**}**

BASIC-24**基础练习 龟兔赛跑预测**

时间限制：1.0s   内存限制：512.0MB

关键字：数组 模拟

问题描述

　　话说这个世界上有各种各样的兔子和乌龟，但是研究发现，所有的兔子和乌龟都有一个共同的特点——喜欢赛跑。于是世界上各个角落都不断在发生着乌龟和兔子的比赛，小华对此很感兴趣，于是决定研究不同兔子和乌龟的赛跑。他发现，兔子虽然跑比乌龟快，但它们有众所周知的毛病——骄傲且懒惰，于是在与乌龟的比赛中，一旦任一秒结束后兔子发现自己领先t米或以上，它们就会停下来休息s秒。对于不同的兔子，t，s的数值是不同的，但是所有的乌龟却是一致——它们不到终点决不停止。  
　　然而有些比赛相当漫长，全程观看会耗费大量时间，而小华发现只要在每场比赛开始后记录下兔子和乌龟的数据——兔子的速度v1（表示每秒兔子能跑v1米），乌龟的速度v2，以及兔子对应的t，s值，以及赛道的长度l——就能预测出比赛的结果。但是小华很懒，不想通过手工计算推测出比赛的结果，于是他找到了你——清华大学计算机系的高才生——请求帮助，请你写一个程序，对于输入的一场比赛的数据v1，v2，t，s，l，预测该场比赛的结果。

输入格式

　　输入只有一行，包含用空格隔开的五个正整数v1，v2，t，s，l，其中(v1,v2<=100;t<=300;s<=10;l<=10000且为v1,v2的公倍数)

输出格式

　　输出包含两行，第一行输出比赛结果——一个大写字母“T”或“R”或“D”，分别表示乌龟获胜，兔子获胜，或者两者同时到达终点。  
　　第二行输出一个正整数，表示获胜者（或者双方同时）到达终点所耗费的时间（秒数）。

样例输入

10 5 5 2 20

样例输出

D  
4

样例输入

10 5 5 1 20

样例输出

R  
3

样例输入

10 5 5 3 20

样例输出

T  
4

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int v1,v2,t,s,L,L1=0,L2=0,ans=0,i,j;**

**bool bk=true;**

**scanf("%d%d%d%d%d",&v1,&v2,&t,&s,&L);**

**while(1)**

**{**

**if(L1-L2>=t&&L1<L&&L2<L)**

**{**

**for(i=1;i<=s;i++)**

**{**

**if(L1<L&&L2<L)**

**{**

**L2+=v2;**

**ans++;**

**}**

**}**

**}**

**else**

**{**

**L2+=v2;**

**L1+=v1;**

**ans++;**

**}**

**if(L1>=L) break;**

**if(L2>=L) break;**

**}**

**if(L1>=L&&L2>=L){printf("D\n%d\n",ans);return 0;}**

**if(L1>=L) {printf("R\n%d\n",ans); return 0;}**

**if(L2>=L) {printf("T\n%d\n",ans); return 0;}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int v1,v2,t,s,l,s1=0,s2=0,i=0;**

**scanf("%d %d %d %d %d",&v1,&v2,&t,&s,&l);**

**while(s1<l&&s2<l)**

**{**

**s1+=v1;**

**s2+=v2;**

**i++;**

**if(s1==l||s2==l)break;**

**if(s1-s2>=t)s1-=v1\*s;**

**}**

**if(s1>s2) printf("R\n");**

**else if(s2>s1) printf("T\n");**

**else printf("D\n");**

**printf("%d",i);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String[] args)throws Exception {**

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

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

**int v1=Integer.parseInt(str[0]);**

**int v2=Integer.parseInt(str[1]);**

**int t=Integer.parseInt(str[2]);**

**int s=Integer.parseInt(str[3]);**

**int l=Integer.parseInt(str[4]);**

**int x=0,y=0,time=0;**

**for(int i=1;i<=l;i++){**

**if(x-y<t){**

**x+=v1;**

**y+=v2;**

**}**

**else{**

**for(int j=1;j<=s;j++){**

**i=i+1;**

**y+=v2;**

**if(y==l)**

**break;**

**}**

**i=i-1;**

**}**

**if(x==l||y==l){**

**time=i;**

**break;**

**}**

**}**

**if(y==l&&x<l)**

**System.out.println("T");**

**if(x==l&&y<l)**

**System.out.println("R");**

**if(y==l&&x==l)**

**System.out.println("D");**

**System.out.println(time);**

**}**

**}**

BASIC-23**基础练习 芯片测试**

时间限制：1.0s   内存限制：512.0MB

关键字：算法基础 统计 二维数组

问题描述

　　有n（2≤n≤20）块芯片，有好有坏，已知好芯片比坏芯片多。  
　　每个芯片都能用来测试其他芯片。用好芯片测试其他芯片时，能正确给出被测试芯片是好还是坏。而用坏芯片测试其他芯片时，会随机给出好或是坏的测试结果（即此结果与被测试芯片实际的好坏无关）。  
　　给出所有芯片的测试结果，问哪些芯片是好芯片。

输入格式

　　输入数据第一行为一个整数n，表示芯片个数。  
　　第二行到第n+1行为n\*n的一张表，每行n个数据。表中的每个数据为0或1，在这n行中的第i行第j列（1≤i, j≤n）的数据表示用第i块芯片测试第j块芯片时得到的测试结果，1表示好，0表示坏，i=j时一律为1（并不表示该芯片对本身的测试结果。芯片不能对本身进行测试）。

输出格式

　　按从小到大的顺序输出所有好芯片的编号

样例输入

3  
1 0 1  
0 1 0  
1 0 1

样例输出

1 3

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cstring>**

**using namespace std;**

**bool a[25][25];**

**bool v[25];**

**int n;**

**bool dfs(int k)**

**{**

**if (k==n)**

**{**

**int sum=0;**

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

**if (v[i])**

**sum++;**

**if (sum>n-sum)**

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

**if (v[i])**

**printf("%d ",i);**

**return true;**

**}**

**if (v[k]==true)**

**{**

**int len=0,s[25];**

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

**if (!a[k][i] && v[i])**

**{**

**s[++len]=i;**

**v[i]=false;**

**}**

**if (dfs(k+1)) return true;**

**for (int i=1;i<=len;i++)**

**v[s[i]]=true;**

**}**

**if (dfs(k+1)) return true;**

**}**

**int main()**

**{**

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

**memset(v,true,sizeof(v));**

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

**for (int j=1;j<=n;j++)**

**{**

**int c;**

**scanf("%d",&c);**

**if (c) a[i][j]=1; else a[i][j]=0;**

**}**

**dfs(1);**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**int a[50][50];int i,j,n,s;scanf("%d",&n);**

**for(j=0;j<n;j++)for(i=0;i<n;i++)scanf("%d",&a[i][j]);**

**for(j=0;j<n;j++){s=0;for(i=0;i<n;i++)s=s+a[j][i];if(s>n/2.0)printf("%d ",j+1);}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String[] args)throws Exception {**

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

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

**int a[][]=new int[n][n];**

**int b[]=new int[n\*n];**

**int i=0,j=0;**

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

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

**for(j=0;j<n;j++){**

**a[i][j]=Integer.parseInt(s[j]);**

**if(a[i][j]==0)**

**b[j]=b[j]+1;**

**}**

**}**

**for(i=0;i<n;i++)**

**if(b[i]<=n/2)**

**System.out.print(i+1+" ");**

**}**

**}**

BASIC-22**基础练习 FJ的字符串**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 递归

问题描述

　　FJ在沙盘上写了这样一些字符串：  
　　A1 = “A”  
　　A2 = “ABA”  
　　A3 = “ABACABA”  
　　A4 = “ABACABADABACABA”  
　　… …  
　　你能找出其中的规律并写所有的数列AN吗？

输入格式

　　仅有一个数：N ≤ 26。

输出格式

　　请输出相应的字符串AN，以一个换行符结束。输出中不得含有多余的空格或换行、回车符。

样例输入

3

样例输出

ABACABA

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**using namespace std;**

**void dfs(int k,int p)**

**{**

**if (k==1)**

**{**

**printf("%c",p+'A');**

**return;**

**}**

**dfs(k/2,p-1);dfs(1,p);dfs(k/2,p-1);**

**}**

**int main()**

**{**

**int n;**

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

**int sum=1;**

**n--;**

**for (int i=1;i<=n;i++) sum=sum\*2+1;**

**dfs(sum,n);**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**void f(int a)**

**{**

**if(a==0)printf("%c",'A');**

**else**

**{**

**f(a-1);**

**printf("%c",a+'A');**

**f(a-1);**

**}**

**}**

**int main()**

**{**

**int a;scanf("%d",&a);f(a-1);printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

**/\*\***

**\* @param args**

**\* @throws IOException**

**\*/**

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

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

**String s=bf.readLine();**

**int n=Integer.parseInt(s);**

**String a="A";**

**f(n,a,1);**

**}**

**private static void f(int n, String a, int i) {**

**if(n==i){**

**System.out.println(a);**

**return;**

**}**

**else {**

**a=a+String.valueOf((char)(i+65))+a;**

**if(i==n){**

**System.out.println(a);**

**return;**

**}**

**}**

**//while(i<n)**

**f(n,a,i+1);**

**}**

**}**

BASIC-21**基础练习 Sine之舞**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 递归 递推

问题描述

　　最近FJ为他的奶牛们开设了数学分析课，FJ知道若要学好这门课，必须有一个好的三角函数基本功。所以他准备和奶牛们做一个“Sine之舞”的游戏，寓教于乐，提高奶牛们的计算能力。  
　　不妨设  
　　An=sin(1–sin(2+sin(3–sin(4+...sin(n))...)  
　　Sn=(...(A1+n)A2+n-1)A3+...+2)An+1  
　　FJ想让奶牛们计算Sn的值，请你帮助FJ打印出Sn的完整表达式，以方便奶牛们做题。

输入格式

　　仅有一个数：N<201。

输出格式

　　请输出相应的表达式Sn，以一个换行符结束。输出中不得含有多余的空格或换行、回车符。

样例输入

3

样例输出

((sin(1)+3)sin(1–sin(2))+2)sin(1–sin(2+sin(3)))+1

**参考代码：**

**C++**

**#include<stdio.h>**

**void An\_Output(int n, int t)**

**{**

**if(n == t)**

**{**

**printf("sin(%d)", t);**

**return ;**

**}**

**char c;**

**c = t % 2 == 1 ? '+' : '-';**

**printf("sin(%d%c", t, c);**

**An\_Output(n, ++t);**

**printf(")");**

**}**

**void Sn\_Output(int n, int t)**

**{**

**//　Sn=(...(A1+n)A2+n-1)A3+...+2)An+1**

**if(n == t)**

**{**

**return ;**

**}**

**printf("(");**

**Sn\_Output(n, t+1);**

**if(t != n - 1)**

**{**

**printf(")");**

**}**

**An\_Output(n - t, 1);**

**printf("+%d", t+1);**

**}**

**int main()**

**{**

**int n;**

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

**Sn\_Output(n, 1);**

**if(n!=1)**

**printf(")");**

**An\_Output(n, 1);**

**printf("+1\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**void printA(int n, int k)**

**{**

**if (n == k)**

**printf("sin(%d)", n);**

**else**

**{**

**printf("sin(%d", n);**

**printf(n % 2 == 0 ? "+" : "-");**

**printA(n + 1, k);**

**printf(")");**

**}**

**}**

**void printS(int n, int k)**

**{**

**if (n == 1)**

**{**

**printA(1, n);**

**printf("+%d", k - n);**

**}**

**else**

**{**

**printf("(");**

**printS(n - 1, k);**

**printf(")");**

**printA(1, n);**

**printf("+%d", k - n);**

**}**

**}**

**int main()**

**{**

**int N;**

**scanf("%d", &N);**

**printS(N, N + 1);**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**int n = scanner.nextInt();**

**String string = f(n, 1, 1);**

**System.out.println(string);**

**}**

**private static String f1(int n, int i, int j) {**

**if (i == n)**

**return "sin(" + j + ")";**

**if (i % 2 == 0)**

**return "sin(" + i + "+" + f1(n, i + 1, j + 1) + ")";**

**else**

**return "sin(" + i + "-" + f1(n, i + 1, j + 1) + ")";**

**}**

**private static String f(int n, int i, int j) {**

**if (i == n)**

**return f1(n, i, j) + "+" + i;**

**return "(" + f(n, i + 1, j) + ")" + f1(n, i, j) + "+" + i;**

**}**

**}**

BASIC-20**基础练习 数的读法**

时间限制：1.0s   内存限制：512.0MB

关键字：判断 函数

问题描述

　　Tom教授正在给研究生讲授一门关于基因的课程，有一件事情让他颇为头疼：一条染色体上有成千上万个碱基对，它们从0开始编号，到几百万，几千万，甚至上亿。  
　　比如说，在对学生讲解第1234567009号位置上的碱基时，光看着数字是很难准确的念出来的。  
　　所以，他迫切地需要一个系统，然后当他输入12 3456 7009时，会给出相应的念法：  
　　十二亿三千四百五十六万七千零九  
　　用汉语拼音表示为  
　　shi er yi san qian si bai wu shi liu wan qi qian ling jiu  
　　这样他只需要照着念就可以了。  
　　你的任务是帮他设计这样一个系统：给定一个阿拉伯数字串，你帮他按照中文读写的规范转为汉语拼音字串，相邻的两个音节用一个空格符格开。  
　　注意必须严格按照规范，比如说“10010”读作“yi wan ling yi shi”而不是“yi wan ling shi”，“100000”读作“shi wan”而不是“yi shi wan”，“2000”读作“er qian”而不是“liang qian”。

输入格式

　　有一个数字串，数值大小不超过2,000,000,000。

输出格式

　　是一个由小写英文字母，逗号和空格组成的字符串，表示该数的英文读法。

样例输入

1234567009

样例输出

shi er yi san qian si bai wu shi liu wan qi qian ling jiu

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cstring>**

**#include<cstdlib>**

**using namespace std;**

**char df[][10]={"ling","yi","er","san","si","wu","liu","qi","ba","jiu"};**

**char s[15];**

**int main()**

**{**

**scanf("%s",s);**

**int lens=strlen(s);**

**bool bk=false;**

**for (int i=0;i<lens;i++)**

**{**

**int p,lendf;**

**p=s[i]-'0';**

**if (p!=0)**

**{**

**bk=false;**

**lendf=strlen(df[p]);**

**if (s[i-1]-'0'==0)**

**printf("ling ");**

**if ((lens-i)%4==2 && p==1 /\*&& s[i-1]-'0'==0 && s[i-2]-'0'==0\*/ && i==0)**

**{**

**printf("shi ");**

**continue;**

**}**

**for (int j=0;j<lendf;j++)**

**printf("%c",df[p][j]);**

**printf(" ");**

**if ((lens-i)%4==2) printf("shi ");**

**if ((lens-i)%4==3) printf("bai ");**

**if ((lens-i)%4==0) printf("qian ");**

**}**

**if ((lens-i)%4==1)**

**{**

**if ((lens-i)/4==2)**

**{**

**bk=true;**

**printf("yi ");**

**}**

**if (bk==false && (lens-i)/4==1) printf("wan ");**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char a[100];int i,j,k,l;**

**char b[20][10]={"ling","yi","er","san","si","wu","liu","qi","ba","jiu"};**

**char c[20][10]={"","","shi","bai","qian","wan","shi","bai","qian","yi","shi"};**

**gets(a);**

**l=strlen(a);**

**for(i=0;i<l;i++){**

**j=a[i]-48;**

**if(j==0){if(i<l-1)if(a[i+1]!=48)printf("%s ",b[j]);}**

**else**

**if((l-i==2||l-i==6||l-i==10)&&j==1)printf("%s ",c[l-i]);**

**else if(a[i-1]==48&&j==1)printf("%s ",c[l-i]);**

**else printf("%s %s ",b[j],c[l-i]);**

**}**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

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

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

**String num = in.readLine();**

**char[] ch = num.toCharArray();**

**getRed(ch,num);**

**}**

**public static void getRed(char[] ch,String str){**

**int numL01 = ch.length;**

**String[] numR01 = new String[numL01];**

**for(int a = 0;a < numL01;a++){**

**numR01[a] = getNum(ch[a] + "");**

**}**

**int numL02 = 2 \* numL01 - 1;**

**String[] numR02 = new String[numL02];**

**numR02[0] = numR01[0];**

**int i01 = 10 - numL01;**

**int i02 = 1;**

**for(int a = 1;a < numL02;a++){**

**if((a%2) == 0){**

**numR02[a] = numR01[i02];**

**i02++;**

**}else{**

**numR02[a] = getR(i01);**

**i01++;**

**}**

**}**

**String red = numR02[0];**

**for(int a = 1;a < numL02;a++){**

**if(a%2 == 0){**

**if(red.endsWith("ling ")){**

**if(getT(numR02[a]))red += numR02[a];**

**}else{**

**red += numR02[a];**

**}**

**}else{**

**if(red.endsWith("ling ")){**

**if(getTR(numR02[a]))red = red.substring(0, red.length() - 5) + numR02[a];**

**}else{**

**red += numR02[a];**

**}**

**}**

**}**

**if(red.endsWith("ling "))red = red.substring(0, red.length() - 5);**

**if((str.length() > 8)**

**& (red.startsWith("yi shi "))){**

**red = red.substring(3);**

**}**

**if(red.indexOf(" i wan ")!=-1){**

**red = red.replaceFirst(" i wan ", " yi ");**

**}else if(red.indexOf(" i ")!=-1){**

**red = red.replaceFirst(" i ", " yi ");**

**}**

**if(str.length() == 6){**

**if(red.indexOf("yi shi ")!=-1){**

**red = red.replaceFirst("yi shi ", "shi ");**

**}**

**}**

**System.out.println(red);**

**}**

**public static String getNum(String str){**

**String[] num = {"ling ","yi ","er ","san ","si ","wu ","liu ","qi ","ba ","jiu "};**

**str = num[Integer.parseInt(str)];**

**return str;**

**}**

**public static String getR(int i){**

**String[] red = {"shi ","i ","qian ","bai ","shi ","wan ","qian ","bai ","shi ",""};**

**return red[i];**

**}**

**public static boolean getT(String str){**

**String[] red = {"yi ","er ","san ","si ","wu ","liu ","qi ","ba ","jiu "};**

**boolean num = false;**

**for(int a = 0;a < red.length;a++){**

**if(str.equals(red[a]))num = true;**

**if(str.equals(red[a]))break;**

**}**

**return num;**

**}**

**public static boolean getTR(String str){**

**boolean r = false;**

**if((str.equals("wan ")) | (str.equals("i ")))r = true;**

**return r;**

**}**

**}**

BASIC-19**基础练习 完美的代价**

时间限制：1.0s   内存限制：512.0MB

**关键字：**贪心算法

问题描述

　　回文串，是一种特殊的字符串，它从左往右读和从右往左读是一样的。小龙龙认为回文串才是完美的。现在给你一个串，它不一定是回文的，请你计算最少的交换次数使得该串变成一个完美的回文串。  
　　交换的定义是：交换两个相邻的字符  
　　例如mamad  
　　第一次交换 ad : mamda  
　　第二次交换 md : madma  
　　第三次交换 ma : madam (回文！完美！)

输入格式

　　第一行是一个整数N，表示接下来的字符串的长度(N <= 8000)  
　　第二行是一个字符串，长度为N.只包含小写字母

输出格式

　　如果可能，输出最少的交换次数。  
　　否则输出Impossible

样例输入

5  
mamad

样例输出

3

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**using namespace std;**

**int changes(char s[],char x,int n)**

**{**

**int i,change=0,j,k;**

**for(i=0;i<n/2;i++)**

**{**

**if(s[i]==x)**

**{**

**for(j=i;j<n-i-1;j++)**

**if(s[n-i-1]==s[j])**

**break;**

**change+=j-i;**

**for(k=j;k>i;k--)**

**s[k]=s[k-1];**

**s[i]=s[n-i-1];**

**}**

**else**

**{**

**for(j=n-i-1;j>=i;j--)**

**if(s[i]==s[j])**

**break;**

**change+=n-i-1-j;**

**for(k=j;k<n-i-1;k++) s[k]=s[k+1];**

**s[n-i-1]=s[i];**

**}**

**}**

**return change;**

**}**

**int main()**

**{**

**int n,i,k=0,b[26]={0},j;**

**char y,s[8001]={0};**

**scanf("%d\n",&n);**

**for(i=0;i<n;i++)**

**{**

**scanf("%c",&s[i]);**

**b[s[i]-'a']++;**

**}**

**char x;**

**for(j=0;j<26;j++)**

**{**

**if(b[j]%2!=0)**

**{**

**k++;**

**x=j+'a';**

**}**

**}**

**if(k>=2)**

**printf("Impossible\n");**

**else**

**{**

**printf("%d\n",changes(s,x,n));**

**return 0;**

**}**

**}**

**C**

**#include<stdio.h>**

**int changes(char s[],char x,int n);**

**char x='0';**

**int main(void)**

**{**

**int n,i,k=0,b[26]={0},j;**

**char y,s[8000]={0};**

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

**getchar();**

**for(i=0;i<n;i++)**

**scanf("%c",&s[i]);**

**for(i=0;i<n;i++)**

**{**

**j=s[i]-'a';**

**b[j]++;**

**}**

**for(j=0;j<26;j++)**

**{**

**if(b[j]%2!=0)**

**{**

**k++;**

**x=j+'a';**

**}**

**}**

**if(k>=2)**

**printf("Impossible\n");**

**else**

**printf("%d\n",changes(s,x,n));**

**return 0;**

**}**

**int changes(char s[],char x,int n)**

**{**

**int i,change=0,j,k;**

**for(i=0;i<n/2;i++)**

**{**

**if(s[i]==x)**

**{**

**for(j=i;j<n-i-1;j++)**

**if(s[n-i-1]==s[j])**

**break;**

**change+=j-i;**

**for(k=j;k>i;k--)**

**s[k]=s[k-1];**

**s[i]=s[n-i-1];**

**}**

**else**

**{**

**for(j=n-i-1;j>=i;j--)**

**if(s[i]==s[j])**

**break;**

**change+=n-i-1-j;**

**for(k=j;k<n-i-1;k++)**

**s[k]=s[k+1];**

**s[n-i-1]=s[i];**

**}**

**}**

**return change;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main{**

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

**BufferedReader buf = new BufferedReader(**

**new InputStreamReader(System.in));**

**int n = Integer.parseInt(buf.readLine());**

**String s = buf.readLine();**

**char c[] = s.toCharArray();**

**int count = 0, num = 0;**

**char ch = '0';**

**int i, k = 0, j;**

**int b[] = new int[26];**

**for (i = 0; i < n; i++)**

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

**j = c[i] - 'a';**

**b[j]++;**

**}**

**for (j = 0; j < 26; j++) {**

**if (b[j] % 2 != 0) {**

**k++;**

**ch = (char) (j + 'a');**

**}**

**}**

**if (k >= 2)**

**System.out.println("Impossible");**

**else**

**System.out.println(changes(c, ch, n));**

**}**

**public static int changes(char s[], char x, int n) {**

**int i, change = 0, j, k;**

**for (i = 0; i < n / 2; i++) {**

**if (s[i] == x) {**

**for (j = i; j < n - i - 1; j++)**

**if (s[n - i - 1] == s[j])**

**break;**

**change += j - i;**

**for (k = j; k > i; k--)**

**s[k] = s[k - 1];**

**s[i] = s[n - i - 1];**

**} else {**

**for (j = n - i - 1; j >= i; j--)**

**if (s[i] == s[j])**

**break;**

**change += n - i - 1 - j;**

**for (k = j; k < n - i - 1; k++)**

**s[k] = s[k + 1];**

**s[n - i - 1] = s[i];**

**}**

**}**

**return change;**

**}**

**}**

BASIC-18**基础练习 矩形面积交**

时间限制：1.0s   内存限制：512.0MB

关键字：判断 线段交

问题描述

　　平面上有两个矩形，它们的边平行于直角坐标系的X轴或Y轴。对于每个矩形，我们给出它的一对相对顶点的坐标，请你编程算出两个矩形的交的面积。

输入格式

　　输入仅包含两行，每行描述一个矩形。  
　　在每行中，给出矩形的一对相对顶点的坐标，每个点的坐标都用两个绝对值不超过10^7的实数表示。

输出格式

　　输出仅包含一个实数，为交的面积，保留到小数后两位。

样例输入

1 1 3 3  
2 2 4 4

样例输出

1.00

**参考代码：**

**C++**

**#include <iostream>**

**#include <algorithm>**

**#include <cmath>**

**#include <cstdio>**

**using namespace std;**

**int main()**

**{**

**double x1,x2,y1,y2;**

**double q1,q2,w1,w2;**

**while(cin>>x1>>y1>>x2>>y2>>q1>>w1>>q2>>w2)**

**{**

**double xx=max(min(x1,x2),min(q1,q2));**

**double yy=max(min(y1,y2),min(w1,w2));**

**double xxup=min(max(x1,x2),max(q1,q2));**

**double yyup=min(max(y1,y2),max(w1,w2));**

**if(xxup>xx)**

**printf("%.2f\n",fabs((xx)-(xxup))\*fabs((yy)-(yyup)));**

**else printf("0.00\n");**

**}**

**}**

**C**

**#include <stdio.h>**

**#define max(x,y) ((x)>(y)?(x):(y))**

**#define min(x,y) ((x)<(y)?(x):(y))**

**int main()**

**{**

**double x1,y1,x2,y2;//矩形1**

**double x3,y3,x4,y4;//矩形2**

**double m1,n1;//交集左上角坐标**

**double m2,n2;//交集右下角坐标**

**scanf("%lf%lf%lf%lf",&x1,&y1,&x2,&y2);**

**scanf("%lf%lf%lf%lf",&x3,&y3,&x4,&y4);**

**m1=max(min(x1,x2),min(x3,x4));**

**n1=max(min(y1,y2),min(y3,y4));**

**m2=min(max(x1,x2),max(x3,x4));**

**n2=min(max(y1,y2),max(y3,y4));**

**if(m2>m1&&n2>n1)**

**printf("%.2f\n",(m2-m1)\*(n2-n1));**

**else**

**printf("0.00\n");**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main**

**{**

**public static void main(String[] args)throws IOException**

**{**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**double[][] arr = new double[2][4];**

**String tag1[] = br.readLine().split(" ");**

**String tag2[] = br.readLine().split(" ");**

**for(int b=0;b<4;b++)**

**{**

**arr[0][b] = Double.valueOf(tag1[b]);**

**}**

**for(int b=0;b<4;b++)**

**{**

**arr[1][b] = Double.valueOf(tag2[b]);**

**}**

**double a,b,c,d,e,f,g,h;**

**a = Math.max(arr[0][0],arr[0][2]);**

**b = Math.min(arr[0][0],arr[0][2]);**

**c = Math.max(arr[0][1],arr[0][3]);**

**d = Math.min(arr[0][1],arr[0][3]);**

**e = Math.max(arr[1][0],arr[1][2]);**

**f = Math.min(arr[1][0],arr[1][2]);**

**g = Math.max(arr[1][1],arr[1][3]);**

**h = Math.min(arr[1][1],arr[1][3]);**

**if(a<f || e<b || c<h || g<d)**

**{**

**System.out.println("0.00");**

**}**

**else**

**{**

**double i = (Math.min(a,e)-Math.max(b,f));**

**double j = (Math.min(c,g)-Math.max(d,h));**

**String s = String.format("%.2f",i\*j);**

**System.out.println(s);**

**}**

**}**

**}**

BASIC-17**基础练习 矩阵乘法**

时间限制：1.0s   内存限制：512.0MB

关键字：二维数组 循环 矩阵

问题描述

　　给定一个N阶矩阵A，输出A的M次幂（M是非负整数）  
　　例如：  
　　A =  
　　1 2  
　　3 4  
　　A的2次幂  
　　7 10  
　　15 22

输入格式

　　第一行是一个正整数N、M（1<=N<=30, 0<=M<=5），表示矩阵A的阶数和要求的幂数  
　　接下来N行，每行N个绝对值不超过10的非负整数，描述矩阵A的值

输出格式

　　输出共N行，每行N个整数，表示A的M次幂所对应的矩阵。相邻的数之间用一个空格隔开

样例输入

2 2  
1 2  
3 4

样例输出

7 10  
15 22

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**#include<cstring>**

**using namespace std;**

**int a[101][101];**

**int c[101][101];**

**int ans[101][101];**

**int main()**

**{**

**int i,j,k,l,m,n;**

**scanf("%d%d",&n,&m);**

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

**for(j=1;j<=n;j++)**

**scanf("%d",&a[i][j]);**

**memset(ans,0,sizeof(ans));**

**for(i=1;i<=n;i++) ans[i][i]=1;**

**for(k=1;k<=m;k++)**

**{ memset(c,0,sizeof(c));**

**for(i=1;i<=n;i++)for(j=1;j<=n;j++)for(l=1;l<=n;l++)c[i][j]+=ans[i][l]\*a[l][j];**

**for(i=1;i<=n;i++)for(j=1;j<=n;j++)ans[i][j]=c[i][j];**

**}**

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

**{**

**for(j=1;j<n;j++)printf("%d ",ans[i][j]);**

**printf("%d\n",ans[i][n]);**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**#define N 100**

**int A[N][N],t[N][N],r[N][N];**

**int main()**

**{**

**int n,m,i,j,k;**

**scanf("%d%d",&n,&m);**

**for(i=0;i<n;i++)**

**for(j=0;j<n;j++)**

**scanf("%d",&A[i][j]);**

**for(i=0;i<n;i++)**

**r[i][i]=1;//单位矩阵，如同数的乘法中的1**

**while(m--)**

**{**

**memset(t,0,sizeof(t));**

**for(i=0;i<n;i++)**

**for(j=0;j<n;j++)**

**for(k=0;k<n;k++)**

**t[i][j]+=r[i][k]\*A[k][j];**

**for(i=0;i<n;i++)**

**for(j=0;j<n;j++)**

**r[i][j]=t[i][j];**

**}**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<n-1;j++)**

**printf("%d ",r[i][j]);**

**printf("%d\n",r[i][j]);**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str[] = br.readLine().split(" ");**

**int Fir = Integer.parseInt(str[0]);**

**int Sec = Integer.parseInt(str[1]);**

**int[][] Mat = new int[Fir][Fir];**

**int[][] MatPro = new int[Fir][Fir];**

**for (int row\_1 = 0; row\_1 < Fir; row\_1++) {**

**String input[] = br.readLine().split(" ");**

**for (int row\_2 = 0; row\_2 < Fir; row\_2++) {**

**Mat[row\_1][row\_2] = Integer.parseInt(input[row\_2]);**

**}**

**}**

**func(MatPro, Mat, Fir, Sec);**

**}**

**public static void func(int[][] MatPro, int[][] Mat, int a, int b) {**

**int[][] tag = new int[a][a];**

**for (int x = 0; x < a; x++) {**

**for (int y = 0; y < a; y++) {**

**tag[x][y] = Mat[x][y];**

**}**

**}**

**if (b == 0) {**

**for (int x = 0; x < a; x++) {**

**for (int y = 0; y < a; y++) {**

**if (x == y) {**

**MatPro[x][y] = 1;**

**} else {**

**MatPro[x][y] = 0;**

**}**

**System.out.print(MatPro[x][y] + " ");**

**}**

**System.out.println();**

**}**

**}**

**if (b != 0) {**

**for (int p = 1; p < b; p++) {**

**for (int x = 0; x < a; x++) {**

**for (int y = 0; y < a; y++) {**

**int pro = 0;**

**for (int z = 0; z < a; z++) {**

**pro += Mat[x][z] \* tag[z][y];**

**MatPro[x][y] = pro;**

**}**

**}**

**}**

**for (int x = 0; x < a; x++) {**

**for (int y = 0; y < a; y++) {**

**tag[x][y] = MatPro[x][y];**

**}**

**}**

**}**

**for (int x = 0; x < a; x++) {**

**for (int y = 0; y < a; y++) {**

**System.out.print(tag[x][y] + " ");**

**}**

**System.out.println();**

**}**

**}**

**}**

**}**

BASIC-16**基础练习 分解质因数**

时间限制：1.0s   内存限制：512.0MB

关键字：质数分解 循环

问题描述

　　求出区间[a,b]中所有整数的质因数分解。

输入格式

　　输入两个整数a，b。

输出格式

　　每行输出一个数的分解，形如k=a1\*a2\*a3...(a1<=a2<=a3...，k也是从小到大的)(具体可看样例)

样例输入

3 10

样例输出

3=3  
4=2\*2  
5=5  
6=2\*3  
7=7  
8=2\*2\*2  
9=3\*3  
10=2\*5

提示

　　先筛出所有素数，然后再分解。

数据规模和约定

　　2<=a<=b<=10000

**参考代码：**

**C++**

**#include <stdio.h>**

**#define N 10000//筛法，求1万以内的所有素数**

**int p[N];**

**int main()**

**{**

**int i,j,a,b,t;**

**for(i=2;i<N/2;i++)**

**{**

**if(p[i]) continue;//合数不参加筛法**

**for(j=2\*i;j<=N;j+=i)**

**{**

**p[j]=1;**

**}**

**}**

**scanf("%d%d",&a,&b);**

**for(i=a;i<=b;i++)**

**{**

**t=i;**

**printf("%d=",t);**

**while(t)**

**{**

**for(j=2;j\*j<=t;j++)**

**{**

**if(!p[j]&&!(t%j))**

**{**

**printf("%d\*",j);**

**t/=j;**

**break;**

**}**

**}**

**if(j\*j>t)**

**{**

**printf("%d",t);**

**break;**

**}**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**long int b,i,k,m,n,w = 0;**

**scanf("%ld%ld",&m,&n);**

**for(i = m;i<=n;i++)**

**{**

**printf("%ld=",i);**

**b = i;k = 2;**

**while(k<=sqrt(i))**

**{**

**if(b%k==0)**

**{**

**b = b/k;**

**if(b>1)**

**{**

**printf("%ld\*",k);continue;**

**}**

**if(b==1) printf("%ld\n",k);**

**}**

**k++;**

**}**

**if(b>1&&b<i) printf("%ld\n",b);**

**if(b==i)**

**{**

**printf("%d\n",i);w++;**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main{**

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

**BufferedReader buf=new BufferedReader(new InputStreamReader(System.in));**

**String s=buf.readLine();**

**String str[]=s.split(" ");**

**int i=Integer.parseInt(str[0]);**

**int n=Integer.parseInt(str[1]);**

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

**StringBuffer b=new StringBuffer();**

**b.append(j).append("=");**

**int x=j;**

**int y=(int)Math.sqrt(x);**

**if(y<2){**

**b.append(j);**

**}**

**else{**

**for(int k=2;k<=y;k++){**

**if(x%k==0){**

**b.append(k).append("\*");**

**x=x/k;**

**y=(int)Math.sqrt(x);**

**k=1;**

**}**

**}**

**b.append(x);**

**}**

**System.out.println(b.toString());**

**}**

**}**

**}**

BASIC-15**基础练习 字符串对比**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 大小写

问题描述

　　给定两个仅由大写字母或小写字母组成的字符串(长度介于1到10之间)，它们之间的关系是以下4中情况之一：  
　　1：两个字符串长度不等。比如 Beijing 和 Hebei  
　　2：两个字符串不仅长度相等，而且相应位置上的字符完全一致(区分大小写)，比如 Beijing 和 Beijing  
　　3：两个字符串长度相等，相应位置上的字符仅在不区分大小写的前提下才能达到完全一致（也就是说，它并不满足情况2）。比如 beijing 和 BEIjing  
　　4：两个字符串长度相等，但是即使是不区分大小写也不能使这两个字符串一致。比如 Beijing 和 Nanjing  
　　编程判断输入的两个字符串之间的关系属于这四类中的哪一类，给出所属的类的编号。

输入格式

　　包括两行，每行都是一个字符串

输出格式

　　仅有一个数字，表明这两个字符串的关系编号

样例输入

BEIjing  
beiJing

样例输出

3

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cstring>**

**#include<cmath>**

**using namespace std;**

**char A[15],B[15];**

**int main()**

**{**

**scanf("%s",A);**

**scanf("%s",B);**

**int a=strlen(A);**

**int b=strlen(B);**

**int count=0;**

**if(a!=b) //长度不等**

**{**

**printf("1\n");**

**return 0;**

**}**

**//长度相等**

**else**

**{**

**for(int i=0;i<a;i++)**

**{**

**if((A[i]!=B[i]))**

**{**

**if(abs(A[i]-B[i])!=32)**

**{**

**printf("4\n");**

**return 0;**

**}**

**else**

**{**

**++count;**

**continue;**

**}**

**}**

**}**

**if((count==0))**

**printf("2\n");**

**else**

**printf("3\n");**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char a[10],b[10],i,n,l=2;**

**gets(a);gets(b);**

**n=strlen(a);**

**if(strlen(b)!=n)l=1;**

**else**

**{**

**for(i=0;i<n;i++)**

**if(a[i]==b[i]||a[i]==b[i]+32||a[i]+32==b[i])**

**if(a[i]!=b[i])l=3;**

**else ;else {l=4;break;}**

**}**

**printf("%d",l);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String args[])throws Exception{**

**BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));**

**String str1= null;String str2 = null;**

**str1 = buf.readLine();**

**str2 = buf.readLine();**

**if(str1.length()==str2.length()){**

**if(str1.equals(str2)){**

**System.out.println(2);**

**}else if(str1.equalsIgnoreCase(str2)){**

**System.out.println(3);**

**}else{**

**System.out.println(4);**

**}**

**}else{**

**System.out.println(1);**

**}**

**}**

**}**

ALGO-95**算法训练 2的次幂表示**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　任何一个正整数都可以用2进制表示，例如：137的2进制表示为10001001。  
　　将这种2进制表示写成2的次幂的和的形式，令次幂高的排在前面，可得到如下表达式：137=2^7+2^3+2^0  
　　现在约定幂次用括号来表示，即a^b表示为a（b）  
　　此时，137可表示为：2（7）+2（3）+2（0）  
　　进一步：7=2^2+2+2^0 （2^1用2表示）  
　　3=2+2^0  
　　所以最后137可表示为：2（2（2）+2+2（0））+2（2+2（0））+2（0）  
　　又如：1315=2^10+2^8+2^5+2+1  
　　所以1315最后可表示为：  
　　2（2（2+2（0））+2）+2（2（2+2（0）））+2（2（2）+2（0））+2+2（0）

输入格式

　　正整数（1<=n<=20000）

输出格式

　　符合约定的n的0，2表示（在表示中不能有空格）

样例输入

137

样例输出

2(2(2)+2+2(0))+2(2+2(0))+2(0)

样例输入

1315

样例输出

2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)  
  
提示  
用递归实现会比较简单，可以一边递归一边输出

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**//递归实现 思路是先转换成二进制**

**int fun(int n)**

**{**

**int i=0;**

**int a[20]={0};**

**int m=n;**

**while(m)**

**{**

**a[i]=m%2;**

**m/=2;**

**i++;**

**}**

**for(int j=i-1;j>=0;j--)//高位到低位排列 但是要注意每位的权改变**

**{**

**if(a[j]==1)**

**{**

**//若是最后一个1 则之后不要加号**

**int flag=1;**

**for(int k=j-1;k>=0;k--)**

**{**

**if(a[k]==1)**

**{**

**flag=0;**

**break;**

**}**

**}**

**if(flag)//是最后一位**

**{**

**if(j==1)**

**cout<<"2";**

**else**

**{**

**if(j==0)**

**cout<<"2("<<j<<")";**

**else**

**{**

**cout<<"2(";**

**fun(j);**

**cout<<")";**

**}**

**}**

**}**

**else//不是最后一位**

**{**

**if(j==1)**

**cout<<"2+";**

**else**

**{**

**if(j==0)**

**cout<<"2("<<j<<")+";**

**else**

**{**

**cout<<"2(";**

**fun(j);**

**cout<<")+";**

**}**

**}**

**}**

**}**

**}**

**return 0;**

**}**

**int main()**

**{**

**int n;**

**cin>>n;**

**fun(n);**

**cout<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int l=0;**

**char temp[1000]={0};**

**void show(int n)**

**{**

**if(n==0) {temp[l]='0';l++;return ;}**

**if(n==2) { temp[l]='2',l++;return ; }**

**int a[15]={0},i=0,j;**

**while(n!=0)**

**{**

**a[i]=n%2;**

**n/=2;**

**i++;**

**}**

**for(j=i-1;j>=0;j--)**

**if(a[j]==1)**

**{**

**if(j==1)**

**{**

**if(temp[l-1]==')' || temp[l-1]=='2' ) {temp[l]='+';l++;}**

**temp[l]='2';l++;**

**}**

**else**

**{**

**if(temp[l-1]==')' || temp[l-1]=='2' ) {temp[l]='+';l++;}**

**temp[l]='2';l++;**

**temp[l]='(';l++;**

**show(j);**

**temp[l]=')';l++;**

**}**

**}**

**}**

**int main()**

**{**

**int n;**

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

**show(n);**

**printf("%s",temp);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**/\*\***

**\* @param args**

**\*/**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Scanner sc=new Scanner(System.in);**

**int input=sc.nextInt();**

**System.out.println(f(input));**

**}**

**public static String f(int a){**

**String s=null;**

**String result="";**

**if(a==1)**

**return "2";**

**else if(a==2)**

**return "2(2)";**

**else if(a==0)**

**return "2(0)";**

**else{**

**s= Integer.toBinaryString(a);**

**int len=s.length();**

**for(int i=0;i<len;i++){**

**if(s.charAt(i)=='1')**

**{**

**if(len-i-1>2)**

**{**

**if(result=="")**

**result=result+ "2("+f(len-i-1)+")";**

**else**

**result=result+ "+2("+f(len-i-1)+")";**

**}**

**else if((len-i-1)<=2&&(len-i-1)>=0)**

**{**

**if(result=="")**

**{**

**result=f(len-i-1);**

**}**

**else**

**result=result+"+"+f(len-i-1);**

**}**

**}**

**}**

**return result;**

**}**

**}**

**}**

ALGO-92**算法训练 前缀表达式**

时间限制：1.0s   内存限制：512.0MB

关键字：字符操作 数学知识

问题描述

　　编写一个程序，以字符串方式输入一个前缀表达式，然后计算它的值。输入格式为：“运算符 对象1 对象2”，其中，运算符为“+”（加法）、“-”（减法）、“\*”（乘法）或“/”（除法），运算对象为不超过10的整数，它们之间用一个空格隔开。要求：对于加、减、乘、除这四种运算，分别设计相应的函数来实现。  
　　输入格式：输入只有一行，即一个前缀表达式字符串。  
　　输出格式：输出相应的计算结果（如果是除法，直接采用c语言的“/”运算符，结果为整数）。  
　　输入输出样例

样例输入

+ 5 2

样例输出

7

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**char c;**

**int a,b;**

**cin>>c>>a>>b;**

**int res;**

**if(c=='+')res=a+b;**

**else if(c=='-')res=a-b;**

**else if(c=='\*')res=a\*b;**

**else res=a/b;**

**cout<<res;**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{ int a[2];**

**int i,j;**

**char c=getchar();**

**for(i=0;i<2;i++)**

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

**if(c=='+')**

**j=a[0]+a[1];**

**else if(c=='-')**

**j=a[0]-a[1];**

**else if(c=='\*')**

**j=a[0]\*a[1];**

**else if(c=='/')**

**j=a[0]/a[1];**

**printf("%d",j);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

**public static void input() throws IOException{**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**String str=br.readLine();**

**String []arr=str.split(" ");**

**String s=arr[0];**

**int num1=Integer.parseInt(arr[1]);**

**int num2=Integer.parseInt(arr[2]);**

**switch (s) {**

**case "+":**

**System.out.println(add(num1,num2));**

**break;**

**case "-":**

**System.out.println(cut(num1,num2));**

**break;**

**case "\*":**

**System.out.println(mul(num1,num2));**

**break;**

**case "/":**

**System.out.println(div(num1,num2));**

**break;**

**default:**

**break;**

**}**

**}**

**public static int add(int num1,int num2){**

**return num1+num2;**

**}**

**public static int cut(int num1,int num2){**

**return num1-num2;**

**}**

**public static int mul(int num1,int num2){**

**return num1\*num2;**

**}**

**public static int div(int num1,int num2){**

**return num1/num2;**

**}**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**try {**

**input();**

**} catch (IOException e) {**

**// TODO Auto-generated catch block**

**e.printStackTrace();**

**}**

**}**

**}**

ALGO-91**算法训练 Anagrams问题**

时间限制：1.0s   内存限制：512.0MB

关键字：数组运算 字符操作

问题描述

　　Anagrams指的是具有如下特性的两个单词：在这两个单词当中，每一个英文字母（不区分大小写）所出现的次数都是相同的。例如，“Unclear”和“Nuclear”、“Rimon”和“MinOR”都是Anagrams。编写一个程序，输入两个单词，然后判断一下，这两个单词是否是Anagrams。每一个单词的长度不会超过80个字符，而且是大小写无关的。  
　　输入格式：输入有两行，分别为两个单词。  
　　输出格式：输出只有一个字母Y或N，分别表示Yes和No。  
　　输入输出样例

样例输入

Unclear  
Nuclear

样例输出

Y

**参考代码：**

**C++**

**#include <iostream>**

**#include <cmath>**

**#include <string>**

**#include <windows.h>**

**#include <stack>**

**#include <vector>**

**#include <iomanip>**

**#include <stack>**

**#include <set>**

**#include <map>**

**using namespace std;**

**char GetCapital(char c)**

**{**

**if(c<='Z')**

**return c;**

**else**

**return c-('a'-'A');**

**}**

**int main(int argc, char\*\* argv) {**

**map<char,int> a,b;**

**string t;**

**cin>>t;**

**for(int i=0;i<t.length();i++)**

**a[GetCapital(t[i])]++;**

**cin>>t;**

**for(int i=0;i<t.length();i++)**

**b[GetCapital(t[i])]++;**

**if(a==b)**

**cout<<"Y";**

**else**

**cout<<"N";**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**void sort(char a[],int len)**

**{**

**int i,j,max;**

**for(i=0;i<len;i++)**

**{**

**max=i;**

**for(j=i+1;j<len;j++)**

**if(a[j]>a[max]) max=j;**

**j=a[i];a[i]=a[max];a[max]=j;**

**}**

**}**

**void strtoupper(char a[],int len)**

**{**

**int i;**

**for(i=0;i<len;i++)**

**if(a[i]>='a' && a[i]<='z') a[i]-=32;**

**}**

**int mystrcmp(char a[],int l1,char b[],int l2)**

**{**

**if(l1!=l2) return 0;**

**int i;**

**for(i=0;i<l1;i++)**

**if(a[i]!=b[i]) return 0;**

**return 1;**

**}**

**int mystrlen(char \*p)**

**{**

**int l=0;**

**while(\*p++!=0)**

**l++;**

**return l;**

**}**

**int main()**

**{**

**char s1[1000]={0},s2[1000]={0};**

**int l1,l2;**

**scanf("%s%s",s1,s2);**

**l1=mystrlen(s1);**

**l2=mystrlen(s2);**

**strtoupper(s1,l1);**

**strtoupper(s2,l2);**

**sort(s1,l1);**

**sort(s2,l2);**

**if(mystrcmp(s1,l1,s2,l2)) printf("Y");**

**else printf("N");**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

**public static int quickSort(int low, int high, char arr[]) {**

**char temp = arr[low];**

**while (low < high) {**

**while (low < high && arr[high] >= temp)**

**high--;**

**arr[low] = arr[high];**

**while (low < high && arr[low] <= temp)**

**low++;**

**arr[high] = arr[low];**

**}**

**arr[low] = temp;**

**return low;**

**}**

**public static void sort(int low, int high, char arr[]) {**

**if (low < high) {**

**int middle = quickSort(low, high, arr);**

**sort(low, middle - 1, arr);**

**sort(middle + 1, high, arr);**

**}**

**}**

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

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

**String s1 = bf.readLine();**

**String s2 = bf.readLine();**

**String str1 = s1.toLowerCase();**

**String str2 = s2.toLowerCase();**

**char[] c1 = str1.toCharArray();**

**char[] c2 = str2.toCharArray();**

**sort(0, c1.length - 1, c1);**

**sort(0, c2.length - 1, c2);**

**boolean flag = true;**

**if (c1.length != c2.length)**

**flag = false;**

**else {**

**for (int i = 0; i < c1.length; i++) {**

**if (c1[i] != c2[i])**

**flag = false;**

**}**

**if (flag)**

**System.out.println("Y");**

**else**

**System.out.println("N");**

**}**

**}**

**}**

ALGO-90**算法训练 出现次数最多的整数**

时间限制：1.0s   内存限制：512.0MB

关键字：数组运算

问题描述  
　　编写一个程序，读入一组整数，这组整数是按照从小到大的顺序排列的，它们的个数N也是由用户输入的，最多不会超过20。然后程序将对这个数组进行统计，把出现次数最多的那个数组元素值打印出来。如果有两个元素值出现的次数相同，即并列第一，那么只打印比较小的那个值。  
　　输入格式：第一行是一个整数N，N £ 20；接下来有N行，每一行表示一个整数，并且按照从小到大的顺序排列。  
　　输出格式：输出只有一行，即出现次数最多的那个元素值。  
　　输入输出样例

样例输入

5  
100  
150  
150  
200  
250

样例输出

150

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**using namespace std;**

**int main()**

**{**

**int n;**

**cin>>n;**

**if(n <= 0) return 0;**

**string \*a = new string[n];**

**int i;**

**for(i = 0; i < n; ++i)**

**cin>>a[i];**

**string number = a[0];**

**int count = 1;**

**int flag = 1;**

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

**{**

**if(a[i].compare(a[i - 1]) == 0)**

**flag = flag + 1;**

**if(a[i].compare(a[i - 1]) == 0 || i == n -1)**

**{**

**if(flag > count)**

**{**

**count = flag;**

**number = a[i - 1];**

**}**

**flag = 1;**

**}**

**}**

**cout<<number<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int n,i,j,t,max=1,num=0;**

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

**if(n>0)**

**{**

**int a[n];**

**for(i=0;i<n;i++)**

**scanf("%d",a+i);**

**j=num=a[0];**

**t=1;**

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

**if(a[i]==j)**

**{**

**++t;**

**if(t>max)**

**{**

**max=t;num=a[i];**

**}**

**}**

**else**

**{**

**t=1;**

**j=a[i];**

**}**

**printf("%d",num);**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String[] args) {**

**try{**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in),1);**

**String line=br.readLine();**

**int n=Integer.parseInt(line);**

**if(n>0){**

**int[][] num=new int[2][n];**

**int max=1,maxn=0;**

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

**line=br.readLine();**

**num[0][i]=Integer.parseInt(line);**

**int g=1;**

**for(int j=i-1;j>=0;j--)**

**if(num[0][j]==num[0][i]){**

**g=num[1][j]+1;**

**break;**

**}**

**num[1][i]=g;**

**if(g>max){**

**max=g;**

**maxn=i;**

**}**

**}**

**System.out.println(num[0][maxn]);**

**}else System.out.println("");**

**}catch(Exception e){**

**System.out.println("Message:"+e.getMessage());**

**}**

**}**

**}**

ALGO-87**算法训练 字串统计**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个长度为n的字符串S，还有一个数字L，统计长度大于等于L的出现次数最多的子串（不同的出现可以相交），如果有多个，输出最长的，如果仍然有多个，输出第一次出现最早的。

输入格式

　　第一行一个数字L。  
　　第二行是字符串S。  
　　L大于0，且不超过S的长度。

输出格式

　　一行，题目要求的字符串。  
  
　　输入样例1：  
　　4  
　　bbaabbaaaaa  
  
　　输出样例1：  
　　bbaa  
  
　　输入样例2：  
　　2  
　　bbaabbaaaaa  
  
　　输出样例2：  
　　aa

数据规模和约定

　　n<=60  
　　S中所有字符都是小写英文字母。  
  
　　提示  
　　枚举所有可能的子串，统计出现次数，找出符合条件的那个

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstring>**

**int main()**

**{**

**char s[65],str[65];**

**int max=0,t,n,len;**

**scanf("%d%s",&n,s);**

**len=strlen(s);**

**if(n<=len)**

**{**

**char ss[65],tt[65];**

**for(int i=len;i>=n;i--)**

**{**

**ss[i]=tt[i]='\0';**

**for(int j=0;j<=len-i;j++)**

**{**

**t=1;**

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

**ss[k]=s[k+j];**

**for(int x=j+1;x<=len-i;x++)**

**{**

**for(int y=0;y<i;y++)**

**tt[y]=s[y+x];**

**if(strcmp(ss,tt)==0)**

**t++;**

**}**

**if(t>max)**

**{**

**max=t;**

**strcpy(str,ss);**

**// printf("%s\n",str);**

**}**

**}**

**}**

**printf("%s\n",str);**

**}**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main(){**

**char S[1000],str[1000][1000],temp[100],out[100];**

**int L,i=0,s,otongji=0,ttongji,a,b,c;**

**scanf("%d%c%c",&L,&S[0],&S[0]);**

**while(S[i]!='\n'){**

**scanf("%c",&S[i+1]);**

**i++;**

**}**

**S[i] = '\0';**

**for(s=i+1;L<=s;L++){**

**for(a=0;a<s+1-L;a++){//赋值**

**for(b = 0;b < L;b++){**

**str[a][b]=S[a+b];**

**}**

**str[a][b] = '\0';**

**}**

**for(i = 0;i < a-1;){//比较**

**for(b = 0;b<a;b++){**

**if(str[b][0]!='\0'){**

**for(c = 0;c<L;c++){**

**temp[c]=str[b][c];**

**}**

**temp[c] = '\0';**

**ttongji = 1;**

**i++;**

**str[b][0] = '\0';**

**break;**

**}**

**}**

**for(b++;b<a;b++){**

**if(!strcmp(str[b],temp)){**

**ttongji++;**

**i++;**

**str[b][0] = '\0';**

**}**

**}**

**if(ttongji > otongji||(ttongji==otongji&&strlen(temp)>strlen(out))){**

**strcpy(out,temp);**

**otongji = ttongji;**

**}**

**}**

**}**

**i = 0;**

**while(out[i] != '\0'){**

**printf("%c",out[i]);**

**i++;**

**}**

**getchar();**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args){**

**Scanner sc=new Scanner(System.in);**

**Main a=new Main();**

**int L=sc.nextInt();**

**String str=sc.next();**

**a.childstr(L, str);**

**}**

**public void childstr(int L,String str){**

**String[][] s=new String[60][60];**

**int[][] in=new int[60][60];**

**int max=0;**

**int td = 0;**

**int tr = 0;**

**for(int i=L;i<str.length();i++)**

**for(int j=0;j<=str.length()-i;j++){**

**s[i][j]=str.substring(j, j+i);**

**//System.out.println(i+" "+j+" "+s[i][j]);**

**}**

**for(int i=L;i<str.length();i++)**

**for(int j=0;j<=str.length()-i;j++)**

**for(int k=j;k<=str.length()-i;k++)**

**if(s[i][k].equals(s[i][j])){**

**in[i][j]=in[i][j]+1;**

**//System.out.println(i+" "+j+" "+k+" "+in[i][j]);**

**}**

**for(int i=L;i<=str.length();i++)**

**for(int j=0;j<=str.length()-i;j++){**

**if(in[i][j]>=max&&td!=i){**

**max=in[i][j];**

**td=i;**

**tr=j;**

**}**

**else if(in[i][j]>max){**

**max=in[i][j];**

**td=i;**

**tr=j;**

**}**

**}**

**System.out.print(s[td][tr]);**

**}**

**}**

ALGO-86**算法训练 矩阵乘法**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入两个矩阵，分别是m\*s，s\*n大小。输出两个矩阵相乘的结果。

输入格式

　　第一行，空格隔开的三个正整数m,s,n（均不超过200）。  
　　接下来m行，每行s个空格隔开的整数，表示矩阵A（i，j）。  
　　接下来s行，每行n个空格隔开的整数，表示矩阵B（i，j）。

输出格式

　　m行，每行n个空格隔开的整数，输出相乘後的矩阵C（i，j）的值。

样例输入

2 3 2  
1 0 -1  
1 1 -3  
0 3  
1 2  
3 1

样例输出

-3 2  
-8 2  
  
提示  
矩阵C应该是m行n列，其中C(i,j)等于矩阵A第i行行向量与矩阵B第j列列向量的内积。  
例如样例中C(1,1)=(1,0,-1)\*(0,1,3) = 1 \* 0 +0\*1+(-1)\*3=-3

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<string.h>**

**#include<stdlib.h>**

**#include<ctype.h>**

**int main()**

**{**

**int m,s,n;**

**int sum=0;**

**int i,j,l;**

**int a[200][200],b[200][200],c[200][200];**

**scanf("%d%d%d",&m,&s,&n);**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<s;j++)**

**scanf("%d",&a[i][j]);**

**}**

**for(i=0;i<s;i++)**

**{**

**for(j=0;j<n;j++)**

**scanf("%d",&b[i][j]);**

**}**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<n;j++)**

**{**

**for(l=0;l<s;l++)**

**{**

**sum+=(a[i][l]\*b[l][j]);**

**}**

**c[i][j]=sum;**

**sum=0;**

**}**

**}**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<n;j++)**

**printf("%d ",c[i][j]);**

**printf("\n");**

**}**

**return 0;**

**}**

**C**

**# include<stdio.h>**

**int main(){**

**int m,s,n,i,j,k,a[200][200],b[200][200],c[200][200];**

**scanf("%d%d%d",&m,&s,&n);**

**for(i=1;i<=m;i++){**

**for(j=1;j<=s;j++)**

**scanf("%d",&a[i][j]);**

**}**

**for(i=1;i<=s;i++){**

**for(j=1;j<=n;j++)**

**scanf("%d",&b[i][j]);**

**}**

**for(i=1;i<=m;i++){**

**for(j=1;j<=n;j++)**

**c[i][j]=0;**

**}**

**for(i=1;i<=m;i++){**

**for(j=1;j<=n;j++){**

**for(k=1;k<=s;k++){**

**c[i][j]=c[i][j]+a[i][k]\*b[k][j];**

**}**

**}**

**}**

**for(i=1;i<=m;i++){**

**for(j=1;j<=n;j++)**

**printf("%d ",c[i][j]);**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args)**

**{**

**int m,s,n;**

**Scanner in=new Scanner(System.in);**

**m=in.nextInt();**

**s=in.nextInt();**

**n=in.nextInt();**

**int[][] A=new int[m][s];**

**int[][] B=new int[s][n];**

**for(int i=0;i<m;i++)**

**{**

**for(int j=0;j<s;j++)**

**A[i][j]=in.nextInt();**

**}**

**for(int i=0;i<s;i++)**

**{**

**for(int j=0;j<n;j++)**

**B[i][j]=in.nextInt();**

**}**

**for(int i=0;i<m;i++)**

**{**

**for(int j=0;j<n;j++)**

**{**

**int sum=0;**

**for(int k=0;k<s;k++)**

**{**

**sum+=A[i][k]\*B[k][j];**

**}**

**System.out.print(sum+" ");**

**}**

**System.out.println();**

**}**

**}**

**}**

ALGO-84**算法训练 大小写转换**

时间限制：1.0s   内存限制：512.0MB

关键字：字符操作

问题描述

　　编写一个程序，输入一个字符串（长度不超过20），然后把这个字符串内的每一个字符进行大小写变换，即将大写字母变成小写，小写字母变成大写，然后把这个新的字符串输出。  
　　输入格式：输入一个字符串，而且这个字符串当中只包含英文字母，不包含其他类型的字符，也没有空格。  
　　输出格式：输出经过转换后的字符串。  
　　输入输出样例

样例输入

AeDb

样例输出

aEdB

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main()**

**{**

**string str;**

**cin>>str;**

**unsigned i;**

**for(i=0;i<str.size();i++)**

**{**

**if(str[i]>='A'&&str[i]<='Z')**

**str[i]+=32;**

**else if(str[i]>='a'&&str[i]<='z')**

**str[i]-=32;**

**}**

**cout<<str;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int i;**

**char ch[100];**

**gets(ch);**

**i=0;**

**while(ch[i]!='\0')**

**{**

**if(ch[i]<='z'&&ch[i]>='a')**

**ch[i]-=32;**

**else ch[i]+=32;**

**i++;**

**}**

**puts(ch);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String[] args)throws Exception{**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**String s=br.readLine();**

**int n=s.length();**

**char ch[]=new char[n];**

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

**ch[i]=s.charAt(i);**

**}**

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

**if(ch[i]>='A'&&ch[i]<='Z'){**

**ch[i]+=32;**

**}else{**

**ch[i]-=32;**

**}**

**}**

**String s1=new String(ch);**

**System.out.println(s1);**

**}**

**}**

ALGO-81**算法训练 动态数组使用**

时间限制：1.0s   内存限制：512.0MB

从键盘读入n个整数，使用动态数组存储所读入的整数，并计算它们的和与平均值分别输出。要求尽可能使用函数实现程序代码。平均值为小数的只保留其整数部分。

样例输入

5  
3 4 0 0 2

样例输出

9 1

样例输入

7  
3 2 7 5 2 9 1

样例输出

29 4

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int n,a,sum=0;**

**cin>>n;**

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

**{**

**cin>>a;**

**sum+=a;**

**}**

**cout<<sum<<" "<<sum/n<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int i,n,a[100],b[100],sum=0,avg=0;**

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

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&b[i]);**

**a[i]=b[i];**

**sum=sum+a[i];**

**}**

**avg=sum/n;**

**printf("%d %d\n",sum,avg);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**/\*\***

**\* @author 林梵**

**\*/**

**public static void main(String[] args) {**

**Scanner sc=new Scanner(System.in);**

**int n=sc.nextInt();**

**int sum=0;**

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

**sum+=sc.nextInt();**

**}**

**System.out.println(sum+" "+sum/n);**

**}**

**}**

ALGO-79**算法训练 删除数组零元素**

时间限制：1.0s   内存限制：512.0MB

从键盘读入n个整数放入数组中，编写函数CompactIntegers，删除数组中所有值为0的元素，其后元素向数组首端移动。注意，CompactIntegers函数需要接受数组及其元素个数作为参数，函数返回值应为删除操作执行后数组的新元素个数。输出删除后数组中元素的个数并依次输出数组元素。　　样例输入: （输入格式说明：5为输入数据的个数，3 4 0 0 2 是以空格隔开的5个整数）  
　　5  
　　3 4 0 0 2  
　　样例输出:（输出格式说明：3为非零数据的个数，3 4 2 是以空格隔开的3个非零整数）  
　　3  
　　3 4 2

样例输入

7  
0 0 7 0 0 9 0

样例输出

2  
7 9

样例输入

3  
0 0 0

样例输出

0

**参考代码：**

**C++**

**#include "iostream"**

**using namespace std;**

**int main()**

**{**

**int n;**

**int \*arr;**

**cin>>n;**

**arr = new int[n];**

**int num = 0;**

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

**{**

**cin>>arr[i];**

**if(arr[i] != 0)**

**{**

**++num;**

**}**

**}**

**cout<<num<<endl;**

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

**{**

**if(arr[i] == 0)**

**{**

**continue;**

**}**

**cout<<arr[i]<<" ";**

**}**

**return 0;**

**}**

**C**

**#include "stdio.h"**

**int CompactIntegers(int a[], int len)**

**{**

**int i, j, k;**

**for(k=0; k<len; k++)**

**for(i=0; i<len; i++)**

**{**

**if(a[i] == 0)**

**{**

**for(j=i; j<len-1; j++)**

**a[j] = a[j+1];**

**len--;**

**}**

**}**

**return len;**

**}**

**void print(int a[], int len)**

**{**

**int i;**

**for(i=0; i<len; i++)**

**printf("%d ", a[i]);**

**printf("\n");**

**}**

**int main()**

**{**

**int a[100000];**

**int n;**

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

**int i;**

**for(i=0; i<n; i++)**

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

**int len = CompactIntegers(a, n);**

**if(len>1)**

**{**

**printf("%d\n", len);**

**print(a, len);**

**}**

**else**

**printf("%d", 0);**

**getchar();**

**getchar();**

**getchar();**

**return 0;**

**}**

**JAVA**

**import java.util.ArrayList;**

**import java.util.List;**

**import java.util.Scanner;**

**public class Main {**

**public void input() {**

**Scanner cin =new Scanner(System.in);**

**int n = cin.nextInt();**

**int [] num =new int[n];**

**int [] num1 =new int[n];**

**int j=0;**

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

**num1[i]=cin.nextInt();**

**if(num1[i]!=0){**

**num[j++]=num1[i];**

**}**

**}**

**System.out.println(j);**

**if(j>0){**

**for(int i=0;i<j;i++)**

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

**}**

**System.out.println();**

**}**

**public static void main(String[] args) {**

**new Main().input();**

**}**

**}**

ALGO-53**算法训练 最小乘积(基本型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给两组数，各n个。  
　　请调整每组数的排列顺序，使得两组数据相同下标元素对应相乘，然后相加的和最小。要求程序输出这个最小值。  
　　例如两组数分别为:1 3　　-5和-2 4 1  
  
　　那么对应乘积取和的最小值应为：  
　　(-5) \* 4 + 3 \* (-2) + 1 \* 1 = -25

输入格式

　　第一个行一个数T表示数据组数。后面每组数据，先读入一个n，接下来两行每行n个数，每个数的绝对值小于等于1000。  
　　n<=8,T<=1000

输出格式

　　一个数表示答案。

样例输入

231 3 -5-2 4 151 2 3 4 51 0 1 0 1

样例输出

-256

**参考代码：**

**C++**

**#include<iostream>**

**#include<algorithm>**

**using namespace std;**

**int a[8],b[8];**

**int main()**

**{**

**int T,n;**

**int i,j;**

**cin>>T;**

**while(T--)**

**{**

**int sum=0;**

**cin>>n;**

**for(i=0;i<n;i++)**

**cin>>a[i];**

**for(i=0;i<n;i++)**

**cin>>b[i];**

**sort(a,a+n);**

**sort(b,b+n);**

**for(i=0;i<n;i++)**

**{**

**sum+=a[i]\*b[n-1-i];**

**}**

**cout<<sum<<endl;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**void sort1(int \*a,int n)**

**{**

**int i,j;**

**int tmp;**

**for(i=0;i<n-1;i++)**

**for(j=0;j<n-1-i;j++)**

**if(a[j]>a[j+1])**

**{**

**tmp=a[j];**

**a[j]=a[j+1];**

**a[j+1]=tmp;**

**}**

**}**

**void sort2(int \*a,int n)**

**{**

**int i,j;**

**int tmp;**

**for(i=0;i<n-1;i++)**

**for(j=0;j<n-1-i;j++)**

**if(a[j]<a[j+1])**

**{**

**tmp=a[j];**

**a[j]=a[j+1];**

**a[j+1]=tmp;**

**}**

**}**

**int main(void)**

**{**

**int T;**

**int n,i;**

**int total;**

**int a[8],b[8],c[8];**

**scanf("%d",&T);**

**while(T)**

**{**

**total=0;**

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

**for(i=0;i<n;i++)**

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

**for(i=0;i<n;i++)**

**scanf("%d",&b[i]);**

**sort1(a,n);**

**sort2(b,n);**

**for(i=0;i<n;i++)**

**c[i]=a[i]\*b[i];**

**for(i=0;i<n;i++)**

**total+=c[i];**

**printf("%d\n",total);**

**T--;**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

**public static void sortInt(int[] num){**

**int temp = 0;**

**for(int i=0;i<num.length-1; i++){**

**for(int j=i+1; j<num.length;j++){**

**if(num[i]<num[j]){**

**temp = num[i];**

**num[i] = num[j];**

**num[j] = temp;**

**}**

**}**

**}**

**}**

**public static void main(String[] args) throws Exception {**

**BufferedReader buf = new BufferedReader(new InputStreamReader(System.in));**

**String temp = buf.readLine();**

**int t = Integer.parseInt(temp);**

**for(int k=0; k<t; k++){**

**temp = buf.readLine();**

**Integer n = Integer.parseInt(temp);**

**String[] tempNum = buf.readLine().split(" ");**

**int[] num1 = new int[n];**

**int[] num2 = new int[n];**

**for(int i=0; i<tempNum.length; i++){**

**num1[i] = Integer.parseInt(tempNum[i]);**

**}**

**tempNum = buf.readLine().split(" ");**

**for(int i=0; i<tempNum.length; i++){**

**num2[i] = Integer.parseInt(tempNum[i]);**

**}**

**sortInt(num1);**

**sortInt(num2);**

**int sum = 0;**

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

**sum += num1[i]\*num2[n-1-i];**

**}**

**System.out.println(sum);**

**}**

**}**

**}**

ALGO-51**算法训练 Torry的困惑(基本型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　Torry从小喜爱数学。一天，老师告诉他，像2、3、5、7……这样的数叫做质数。Torry突然想到一个问题，前10、100、1000、10000……个质数的乘积是多少呢？他把这个问题告诉老师。老师愣住了，一时回答不出来。于是Torry求助于会编程的你，请你算出前n个质数的乘积。不过，考虑到你才接触编程不久，Torry只要你算出这个数模上50000的值。

输入格式

　　仅包含一个正整数n，其中n<=100000。

输出格式

　　输出一行，即前n个质数的乘积模50000的值。

样例输入

1

样例输出

2

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int a[100005];**

**int main()**

**{**

**unsigned int i, j, n, cnt = 1, cj = 2;**

**cin >> n;**

**if (n == 1)**

**{**

**cout << 2 << endl;**

**return 0;**

**}**

**a[0] = 2;**

**for (i = 3; i < 2000000; i++)**

**{**

**for (j = 0; j < cnt; j++)**

**{**

**if (a[j] \* a[j] > i)**

**{**

**break;**

**}**

**else**

**{**

**if (!(i % a[j]))**

**{**

**break;**

**}**

**}**

**}**

**if (a[j] \* a[j] > i)**

**{**

**a[cnt++] = i;**

**cj = (cj % 50000) \* (i % 50000) ;**

**if (cnt == n)**

**{**

**break;**

**}**

**}**

**}**

**cout << cj % 50000 << endl;**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int pr[100010];**

**int top;**

**int isPrime(int n)**

**{**

**int i;**

**for(i = 0; i < top; i++)**

**{**

**if(n % pr[i] == 0)**

**{**

**return 0;**

**}**

**}**

**return 1;**

**}**

**int findNextPrime(void)**

**{**

**int n = pr[top - 1] + 1;**

**while(!isPrime(n))**

**{**

**n++;**

**}**

**pr[top++] = n;**

**return n;**

**}**

**int main(void)**

**{**

**int i, n;**

**int result = 2;**

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

**pr[0] = 2;**

**top = 1;**

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

**{**

**int x = findNextPrime();**

**result \*= x;**

**result %= 50000;**

**}**

**printf("%d", result);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Scanner scanner=new Scanner(System.in);**

**int n=scanner.nextInt();**

**int a[]=new int[100000];**

**FindPrims(a, n);**

**int as=1;**

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

**as=as\*a[i]%50000;**

**}**

**System.out.println(as);**

**}**

**public static boolean isPrims(int x) {**

**for (int i = 2; i <= Math.sqrt(x); i++) {**

**if (x%i==0) {**

**return false;**

**}**

**}**

**return true;**

**}**

**public static void FindPrims(int a[], int n) {**

**int x=1;**

**a[0]=2;**

**int i=1;**

**while (i<n) {**

**x+=2;**

**if (isPrims(x)) {**

**a[i]=x;**

**i++;**

**}**

**}**

**}**

**}**

ALGO-49**算法训练 寻找数组中最大值**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　对于给定整数数组a[],寻找其中最大值，并返回下标。

输入格式

　　整数数组a[],数组元素个数小于1等于100。输出数据分作两行：第一行只有一个数，表示数组元素个数；第二行为数组的各个元素。

输出格式

　　输出最大值，及其下标

样例输入

33 2 1

样例输出

3 0

**参考代码：**

**C++**

**#include<iostream>**

**#include<algorithm>**

**#include<cstring>**

**using namespace std;**

**int main(){**

**int n,a[1000],max,ans;**

**cin>>n;**

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

**cin>>a[i];**

**max=a[0];ans=0;**

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

**if(a[i]>max){**

**max=a[i];**

**ans=i;**

**}**

**}**

**cout<<max<<" "<<ans;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int n,i,k,max;**

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

**int a[n];**

**for(i=0;i<n;i++)**

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

**max=a[0];**

**for(i=0;i<n;i++)**

**{**

**if(a[i]>=max)**

**{**

**max=a[i];**

**k=i;**

**}**

**}**

**printf("%d %d",max,k);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**public static void main(String[] args) throws Exception {**

**// TODO Auto-generated method stub**

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

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

**int []a=new int[n];**

**String str=bf.readLine();**

**String []s=new String[n];**

**s=str.split(" ");**

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

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

**}**

**int max=a[0];**

**int index=0;**

**for(int j=0;j<n;j++){**

**if(a[j]>max){**

**max=a[j];**

**index=j;**

**}**

**}**

**System.out.print(max+" "+index);**

**}**

**}**

ALGO-48**算法训练 关联矩阵**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　有一个n个结点m条边的有向图，请输出他的关联矩阵。

输入格式

　　第一行两个整数n、m，表示图中结点和边的数目。n<=100,m<=1000。  
　　接下来m行，每行两个整数a、b，表示图中有(a,b)边。  
　　注意图中可能含有重边，但不会有自环。

输出格式

　　输出该图的关联矩阵，注意请勿改变边和结点的顺序。

样例输入

5 9  
1 2  
3 1  
1 5  
2 5  
2 3  
2 3  
3 2  
4 3  
5 4

样例输出

1 -1 1 0 0 0 0 0 0  
-1 0 0 1 1 1 -1 0 0  
0 1 0 0 -1 -1 1 -1 0  
0 0 0 0 0 0 0 1 -1  
0 0 -1 -1 0 0 0 0 1

**参考代码：**

**C++**

**#include <iostream>**

**#include <cstdio>**

**using namespace std;**

**int ans[101][1001];**

**int main()**

**{**

**int n,m;**

**int a,b;**

**cin>>n>>m;**

**for(int i=1;i<=m;i++)**

**{**

**scanf("%d%d",&a,&b);**

**ans[a][i] = 1;**

**ans[b][i] = -1;**

**}**

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

**{**

**for(int j=1;j<=m-1;j++)**

**{**

**printf("%d ",ans[i][j]);**

**}**

**printf("%d\n",ans[i][m]);**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int i, ii,n,m, a[1000][2];**

**scanf("%d%d", &n, &m);**

**for ( i = 0; i < m; i++)**

**scanf("%d%d", &a[i][0], &a[i][1]);**

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

**{**

**for (ii = 0; ii < m; ii++)**

**{**

**if (i==a[ii][0])**

**{**

**printf("1 ");**

**}**

**else**

**if (i==a[ii][1])**

**{**

**printf("-1 ");**

**}**

**else**

**{**

**printf("0 ");**

**}**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.util.ArrayList;**

**import java.util.Arrays;**

**import java.util.List;**

**import java.util.Scanner;**

**public class Main {**

**public void method1() throws NumberFormatException, IOException{**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String first = br.readLine();**

**String[] nm = first.split(" ");**

**int n = Integer.parseInt(nm[0].trim());**

**int m = Integer.parseInt(nm[1].trim());**

**int a[][] = new int[n][m];**

**for (int i = 0; i < m; i++) {**

**String second = br.readLine();**

**String[] pq = second.split(" ");**

**int p = Integer.parseInt(pq[0].trim());**

**int q = Integer.parseInt(pq[1].trim());**

**a[p-1][i] = 1;**

**a[q-1][i] = -1;**

**}//end for**

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

**for (int j = 0; j < m; j++) {**

**System.out.print(a[i][j]+" ");**

**}**

**if (i!=n-1) {**

**System.out.println();**

**}**

**}**

**}**

**public static void main(String[] argStrings) throws IOException {**

**Main main = new Main();**

**main.method1();**

**}**

**}**

ALGO-42**算法训练 送分啦**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　这题想得分吗？想，请输出“yes”；不想，请输出“no”。

输出格式

　　输出包括一行，为“yes”或“no”。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstring>**

**using namespace std;**

**int main()**

**{**

**printf("yes\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdio.h>**

**int main()**

**{**

**printf("yes\n");**

**return 0;**

**}**

**JAVA**

**public class Main {**

**public static void main(String[] args) {**

**System.out.println("yes");**

**}**

**}**

ALGO-8**算法训练 操作格子**

时间限制：1.0s   内存限制：256.0MB

**关键字：**线段树

问题描述

有n个格子，从左到右放成一排，编号为1-n。

共有m次操作，有3种操作类型：

1.修改一个格子的权值，

2.求连续一段格子权值和，

3.求连续一段格子的最大值。

对于每个2、3操作输出你所求出的结果。

输入格式

第一行2个整数n，m。

接下来一行n个整数表示n个格子的初始权值。

接下来m行，每行3个整数p,x,y，p表示操作类型，p=1时表示修改格子x的权值为y，p=2时表示求区间[x,y]内格子权值和，p=3时表示求区间[x,y]内格子最大的权值。

输出格式

有若干行，行数等于p=2或3的操作总数。

每行1个整数，对应了每个p=2或3操作的结果。

样例输入

4 3  
1 2 3 4  
2 1 3  
1 4 3  
3 1 4

样例输出

6  
3

数据规模与约定

对于20%的数据n <= 100，m <= 200。

对于50%的数据n <= 5000，m <= 5000。

对于100%的数据1 <= n <= 100000，m <= 100000，0 <= 格子权值 <= 10000。

**参考代码：**

**C++**

**// test**

**// 1.cpp**

**/\***

**ID: Firwaless**

**LANG: C++**

**TASK:**

**\*/**

**#include <cstdio>**

**#include <algorithm>**

**struct Tree**

**{**

**int sum, max;**

**};**

**Tree tree[1 << 18];**

**void scan(int &n)**

**{**

**char c;**

**c = getchar();**

**if (c == EOF) {**

**return ;**

**}**

**while (c < '0' || c > '9') {**

**c = getchar();**

**}**

**n = c - '0';**

**while (c = getchar(), c >= '0' && c <= '9') {**

**n \*= 10;**

**n += c - '0';**

**}**

**}**

**void put(int n)**

**{**

**int cnt = 0;**

**char s[16];**

**if (n == 0) {**

**putchar('0');**

**return ;**

**}**

**for( ; n; n /= 10) {**

**s[cnt++] = n % 10 + '0';**

**}**

**while (cnt--) {**

**putchar(s[cnt]);**

**}**

**}**

**void update(int n, int v)**

**{**

**for (n += (1 << 17),tree[n].sum = tree[n].max = v, n >>= 1; n; n >>= 1) {**

**tree[n].max = std::max(tree[n + n].max, tree[n + n + 1].max);**

**tree[n].sum = tree[n + n].sum + tree[n + n + 1].sum;**

**}**

**}**

**int query(int s, int t, int func)**

**{**

**int sum = 0, max = 0;**

**for (s += (1 << 17) - 1, t += (1 << 17) + 1; s ^ t ^ 1; s >>= 1, t >>= 1) {**

**if (~s & 1) {**

**sum += tree[s ^ 1].sum;**

**max = std::max(max, tree[s ^ 1].max);**

**}**

**if (t & 1) {**

**sum += tree[t ^ 1].sum;**

**max = std::max(max, tree[t ^ 1].max);**

**}**

**}**

**return func ? max : sum;**

**}**

**int main()**

**{**

**int n, m, i, a, b, c;**

**scan(n);scan(m);**

**for (i = 1; i <= n; ++i) {**

**scan(a);**

**update(i, a);**

**}**

**while (m--) {**

**scan(c);scan(a);scan(b);**

**c == 1 && (update(a, b), 0);**

**c == 2 && (put(query(a, b, 0)), putchar('\n'), 0);**

**c == 3 && (put(query(a, b, 1)), putchar('\n'), 0);**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define N 100000**

**#define A 1000**

**#define B 100**

**int sum(int\* a, int m, int n)**

**{**

**int i, s = 0;**

**for (i = m; i <= n; i++)**

**s += a[i];**

**return s;**

**}**

**int max(int\* a, int m, int n)**

**{**

**int i, s = a[m];**

**for (i = m + 1; i <= n; i++)**

**if (s < a[i])**

**s = a[i];**

**return s;**

**}**

**int main()**

**{**

**int i, j, k, m, n;**

**int a[100000], b[100000][3], c[A][2] = {0};**

**scanf("%d%d", &n, &m);**

**for (i = 0; i < n; i++)**

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

**for (i = 0; i < m; i++)**

**for (j = 0; j < 3; j++)**

**scanf("%d", &b[i][j]);**

**for (i = 0; i < (n + B - 1) / B; i++)**

**{**

**c[i][0] = c[i][1] = a[i \* B];**

**for (j = i \* B + 1; j < i \* B + B && j < n; j++)**

**{**

**c[i][0] += a[j];**

**if (c[i][1] < a[j])**

**c[i][1] = a[j];**

**}**

**}**

**for (i = 0; i < m; i++)**

**{**

**if (b[i][0] == 1)**

**{**

**c[(b[i][1] - 1) / B][0] += b[i][2] - a[b[i][1] - 1];**

**k = (b[i][1] - 1) / B;**

**if (c[k][1] <= b[i][2])**

**{**

**c[k][1] = b[i][2];**

**}**

**else if (a[b[i][1] - 1] == c[k][1])**

**{**

**a[b[i][1] - 1] = b[i][2];**

**c[k][1] = max(a, k \* B, k \* B + B > n ? n - 1 : k \* B + B - 1);**

**}**

**a[b[i][1] - 1] = b[i][2];**

**}**

**else if (b[i][0] == 2)**

**{**

**int s = 0;**

**b[i][1]--, b[i][2]--;**

**int o = b[i][2] / B - b[i][1] / B;**

**if (o < 2)**

**{**

**s = sum(a, b[i][1], b[i][2]);**

**}**

**else**

**{**

**s = sum(a, b[i][1], (b[i][1] + B) / B \* B - 1);**

**s += sum(a, b[i][2] / B \* B, b[i][2]);**

**for (j = b[i][1] / B + 1; j < b[i][2] / B; j++)**

**s += c[j][0];**

**}**

**printf("%d\n", s);**

**}**

**else if (b[i][0] == 3)**

**{**

**int s = 0, t;**

**b[i][1]--, b[i][2]--;**

**int o = b[i][2] / B - b[i][1] / B;**

**if (o < 2)**

**{**

**s = max(a, b[i][1], b[i][2]);**

**}**

**else**

**{**

**s = max(a, b[i][1], (b[i][1] + B) / B \* B - 1);**

**t = max(a, b[i][2] / B \* B, b[i][2]);**

**if (s < t) s = t;**

**for (j = b[i][1] / B + 1; j < b[i][2] / B; j++)**

**if (s < c[j][1])**

**s = c[j][1];**

**}**

**printf("%d\n", s);**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedInputStream;**

**import java.io.IOException;**

**import java.util.ArrayList;**

**public class Main**

**{**

**private BufferedInputStream bi = new BufferedInputStream(System.in);**

**private ArrayList<Node> nodes = new ArrayList<Node>();**

**private int size = 0;**

**private Node root;**

**public Main() throws IOException**

**{**

**size = readInt();**

**int p=readInt();**

**root = NextNode(1, size);**

**for(; p>0; p--)**

**{**

**switch(readInt())**

**{**

**case 1:p1();break;**

**case 2:p2();break;**

**case 3:p3();**

**}**

**}**

**}**

**private void p1() throws IOException**

**{**

**Node o = nodes.get(readInt() - 1);**

**int c = readInt();**

**int u = c - o.p;**

**o.s = c;**

**o.p = c;**

**o = o.f;**

**while(o != null)**

**{**

**o.s += u;**

**o.p = o.l.p>o.r.p?o.l.p:o.r.p;**

**o = o.f;**

**}**

**}**

**private void p2() throws IOException**

**{**

**int s = readInt();**

**int e = readInt();**

**System.out.println(sum(s, e, 1, size, root));**

**}**

**private int sum(int s, int e, int l, int r, Node n)**

**{**

**if(s == l && e == r)**

**{**

**return n.s;**

**}**

**else**

**{**

**int m = (r+l)/2;**

**if(e <= m)**

**{**

**return sum(s,e,l,m,n.l);**

**}**

**else if(s > m)**

**{**

**return sum(s,e,m+1,r,n.r);**

**}**

**else**

**{**

**return sum(s,m,l,m,n.l) + sum(m+1,e,m+1,r,n.r);**

**}**

**}**

**}**

**private void p3() throws IOException**

**{**

**int s = readInt();**

**int e = readInt();**

**System.out.println(max(s, e, 1, size, root));**

**}**

**private int max(int s, int e, int l, int r, Node n)**

**{**

**if(s == l && e == r)**

**{**

**return n.p;**

**}**

**else**

**{**

**int m = (r+l)/2;**

**if(e <= m)**

**{**

**return max(s,e,l,m,n.l);**

**}**

**else if(s > m)**

**{**

**return max(s,e,m+1,r,n.r);**

**}**

**else**

**{**

**int p1 = max(s,m,l,m,n.l);**

**int p2 = max(m+1,e,m+1,r,n.r);**

**return p1>p2?p1:p2;**

**}**

**}**

**}**

**public Node NextNode(int l, int r) throws IOException**

**{**

**Node n = new Node();**

**if(l == r)**

**{**

**n.p = n.s = readInt();**

**nodes.add(n);**

**return n;**

**}**

**else**

**{**

**n.l = NextNode(l, (l+r)/2);**

**n.r = NextNode((l+r)/2+1, r);**

**n.l.f = n;**

**n.r.f = n;**

**n.p = n.l.p>n.r.p?n.l.p:n.r.p;**

**n.s = n.l.s + n.r.s;**

**return n;**

**}**

**}**

**public static void main(String[] args) throws IOException**

**{**

**new Main();**

**}**

**private class Node**

**{**

**int p,s;**

**Node l,r;**

**Node f;**

**}**

**private int readInt() throws IOException**

**{**

**int ch;**

**while(!isNum(ch = bi.read())){}**

**int sum = ch - 48;**

**while(isNum(ch = bi.read()))**

**{**

**sum = sum \* 10 + ch - 48;**

**}**

**return sum;**

**}**

**private boolean isNum(int ch)**

**{**

**return (ch >47 && ch < 58);**

**}**

**}**

ALGO-7**算法训练 逆序对**

时间限制：1.0s   内存限制：256.0MB

关键字：平衡二叉树

问题描述

Alice是一个让人非常愉跃的人!他总是去学习一些他不懂的问题，然后再想出许多稀奇古怪的题目。这几天，Alice又沉浸在逆序对的快乐当中，他已近学会了如何求逆序对对数，动态维护逆序对对数等等题目，他认为把这些题让你做简直是太没追求了，于是，经过一天的思考和完善，Alice终于拿出了一道他认为差不多的题目：

有一颗2n-1个节点的二叉树，它有恰好n个叶子节点，每个节点上写了一个整数。如果将这棵树的所有叶子节点上的数从左到右写下来，便得到一个序列a[1]…a[n]。现在想让这个序列中的逆序对数量最少，但唯一的操作就是选树上一个非叶子节点，将它的左右两颗子树交换。他可以做任意多次这个操作。求在最优方案下，该序列的逆序对数最少有多少。

Alice自己已近想出了题目的正解，他打算拿来和你分享，他要求你在最短的时间内完成。

输入格式

第一行一个整数n。

下面每行，一个数x。

如果x=0，表示这个节点非叶子节点，递归地向下读入其左孩子和右孩子的信息，如果x≠0，表示这个节点是叶子节点，权值为x。

输出格式

输出一个整数，表示最少有多少逆序对。

样例输入

3  
0  
0  
3  
1  
2

样例输出

1

数据规模与约定

对于20%的数据，n <= 5000。

对于100%的数据，1 <= n <= 200000，0 <= a[i]<2^31。

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<iostream>**

**using namespace std;**

**#define ForD(i,n) for(int i=n;i;i--)**

**#define F (100000007)**

**#define MAXN (2\*200000+10)**

**long long mul(long long a,long long b){return (a\*b)%F;}**

**long long add(long long a,long long b){return (a+b)%F;}**

**long long sub(long long a,long long b){return (a-b+(a-b)/F\*F+F)%F;}**

**int n,root=0;**

**struct node**

**{**

**int fa,ch[2],size,c;**

**node():size(0),c(0){ch[0]=ch[1]=fa=0;}**

**}a[MAXN];**

**void update(int x){a[x].size=a[a[x].ch[0]].size+a[a[x].ch[1]].size+(a[x].c>0);}**

**int tail=0;**

**void pushdown(int x){a[a[x].ch[0]].fa=a[a[x].ch[1]].fa=x;}**

**void build(int &x)**

**{**

**if (!x) x=++tail;**

**scanf("%d",&a[x].c);**

**if (a[x].c==0)**

**{**

**build(a[x].ch[0]);**

**build(a[x].ch[1]);**

**update(x);pushdown(x);**

**}else a[x].size=1;**

**}**

**void rotate(int x)**

**{**

**int y=a[x].fa,z=a[y].fa;**

**bool p=a[y].ch[0]==x;**

**if (z)**

**{**

**if (a[z].ch[0]==y) a[z].ch[0]=x;**

**else a[z].ch[1]=x;**

**}**

**a[x].fa=z,a[y].fa=x;**

**if (a[x].ch[p]) a[a[x].ch[p]].fa=y;**

**a[y].ch[p^1]=a[x].ch[p];**

**a[x].ch[p]=y;**

**update(y);**

**}**

**void splay(int x)**

**{**

**while (a[x].fa)**

**{**

**int y=a[x].fa,z=a[y].fa;**

**if (z)**

**if ((a[y].ch[0]==x)^(a[z].ch[0]==y)) rotate(x);**

**else rotate(y);**

**rotate(x);**

**}**

**update(x);**

**}**

**void ins(long long &tot,int x,int y)**

**{**

**a[x].size++;**

**if (a[y].c<=a[x].c)**

**{**

**if (a[x].ch[0]) ins(tot,a[x].ch[0],y);**

**else a[y].fa=x,splay(a[x].ch[0]=y);**

**}**

**else**

**{**

**tot+=a[a[x].ch[0]].size+(a[x].c>0);**

**if (a[x].ch[1]) ins(tot,a[x].ch[1],y);**

**else a[y].fa=x,splay(a[x].ch[1]=y);**

**}**

**}**

**int q[MAXN],size;**

**void clac(int x,int y)**

**{**

**if (a[y].ch[0]) clac(x,a[y].ch[0]);**

**if (a[y].c) q[++size]=y;**

**if (a[y].ch[1]) clac(x,a[y].ch[1]);**

**}**

**long long merge(bool &lor,int z)**

**{**

**int x=a[z].ch[0],y=a[z].ch[1];**

**if (a[x].size<a[y].size) swap(x,y);**

**a[x].fa=0;a[y].fa=0;q[1]=y;**

**size=0;clac(x,y);**

**long long tot=0;**

**ForD(i,size)**

**{**

**int now=q[i];**

**a[now].ch[0]=a[now].ch[1]=a[now].fa=0;a[now].size=1;**

**ins(tot,x,now);**

**x=now;**

**}**

**a[x].fa=z;**

**a[z].ch[0]=0,a[z].ch[1]=x;**

**return tot;**

**}**

**long long qur(int &x)**

**{**

**if (a[x].c) return 0;**

**else**

**{**

**long long lson=a[a[x].ch[0]].size,rson=a[a[x].ch[1]].size,ls=qur(a[x].ch[0]),rs=qur(a[x].ch[1]);**

**bool lor=0;**

**long long ms=merge(lor,x);**

**return ls+rs+min(lson\*rson-ms,ms);**

**}**

**}**

**int main()**

**{**

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

**build(root);**

**cout<<qur(root)<<endl;**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#define N 200010**

**long long ans = 0;**

**int left[N], right[N];**

**int len[N];**

**int vals[N];**

**int vTop = 1;**

**int lRotate(int rt)**

**{**

**int nRt = right[rt];**

**right[rt] = left[nRt];**

**left[nRt] = rt;**

**len[nRt] = len[rt];**

**len[rt] = len[left[rt]] + len[right[rt]] + 1;**

**return nRt;**

**}**

**int rRotate(int rt)**

**{**

**int nRt = left[rt];**

**left[rt] = right[nRt];**

**right[nRt] = rt;**

**len[nRt] = len[rt];**

**len[rt] = len[left[rt]] + len[right[rt]] + 1;**

**return nRt;**

**}**

**int adjust(int rt, int isLeft)**

**{**

**if(isLeft)**

**{**

**if(len[left[left[rt]]] > len[right[rt]] || len[right[left[rt]]] > len[right[rt]])**

**{**

**if(len[right[left[rt]]] > len[right[rt]])**

**{**

**left[rt] = lRotate(left[rt]);**

**}**

**return rRotate(rt);**

**}**

**}**

**else**

**{**

**if(len[left[right[rt]]] > len[left[rt]] || len[right[right[rt]]] > len[left[rt]])**

**{**

**if(len[left[right[rt]]] > len[left[rt]])**

**{**

**right[rt] = rRotate(right[rt]);**

**}**

**return lRotate(rt);**

**}**

**}**

**return rt;**

**}**

**int insert(int rt, int node)**

**{**

**len[rt]++;**

**if(vals[node] < vals[rt])**

**{**

**if(left[rt] == 0)**

**{**

**left[rt] = node;**

**}**

**else**

**{**

**left[rt] = insert(left[rt], node);**

**}**

**}**

**else**

**{**

**if(right[rt] == 0)**

**{**

**right[rt] = node;**

**}**

**else**

**{**

**right[rt] = insert(right[rt], node);**

**}**

**}**

**return adjust(rt, vals[node] < vals[rt]);**

**}**

**int rank(int rt, int val)**

**{**

**if(rt == 0)**

**{**

**return 0;**

**}**

**else if(val >= vals[rt])**

**{**

**return rank(right[rt], val);**

**}**

**else**

**{**

**return rank(left[rt], val) + 1 + len[right[rt]];**

**}**

**}**

**int merge(int des, int vBegin, int vEnd)**

**{**

**long long ca = 0, cb = 0;**

**int i;**

**for(i = vBegin; i < vEnd; i++)**

**{**

**ca += rank(des, vals[i]);**

**cb += len[des] - rank(des, vals[i] - 1);**

**}**

**ans += ca < cb ? ca : cb;**

**for(i = vBegin; i < vEnd; i++)**

**{**

**left[i] = right[i] = 0;**

**len[i] = 1;**

**des = insert(des, i);**

**}**

**return des;**

**}**

**int buildTree()**

**{**

**int val;**

**scanf("%d", &val);**

**if(val != 0)**

**{**

**left[vTop] = right[vTop] = 0;**

**len[vTop] = 1;**

**vals[vTop] = val;**

**return vTop++;**

**}**

**int ls = vTop;**

**int rlt = buildTree();**

**int rs = vTop;**

**int rrt = buildTree();**

**int re = vTop;**

**if(rs - ls > re - rs)**

**{**

**return merge(rlt, rs, re);**

**}**

**else**

**{**

**return merge(rrt, ls, rs);**

**}**

**}**

**int main(void)**

**{**

**int n;**

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

**buildTree();**

**printf("%I64d", ans);**

**return 0;**

**}**

**JAVA**

ALGO-6**算法训练 安慰奶牛**

时间限制：1.0s   内存限制：256.0MB

关键字：最小生成树

问题描述

Farmer John变得非常懒，他不想再继续维护供奶牛之间供通行的道路。道路被用来连接N个牧场，牧场被连续地编号为1到N。每一个牧场都是一个奶牛的家。FJ计划除去P条道路中尽可能多的道路，但是还要保持牧场之间 的连通性。你首先要决定那些道路是需要保留的N-1条道路。第j条双向道路连接了牧场Sj和Ej(1 <= Sj <= N; 1 <= Ej <= N; Sj != Ej)，而且走完它需要Lj的时间。没有两个牧场是被一条以上的道路所连接。奶牛们非常伤心，因为她们的交通系统被削减了。你需要到每一个奶牛的住处去安慰她们。每次你到达第i个牧场的时候(即使你已经到过)，你必须花去Ci的时间和奶牛交谈。你每个晚上都会在同一个牧场(这是供你选择的)过夜，直到奶牛们都从悲伤中缓过神来。在早上 起来和晚上回去睡觉的时候，你都需要和在你睡觉的牧场的奶牛交谈一次。这样你才能完成你的 交谈任务。假设Farmer John采纳了你的建议，请计算出使所有奶牛都被安慰的最少时间。

输入格式

第1行包含两个整数N和P。

接下来N行，每行包含一个整数Ci。

接下来P行，每行包含三个整数Sj, Ej和Lj。

输出格式

输出一个整数, 所需要的总时间(包含和在你所在的牧场的奶牛的两次谈话时间)。

样例输入

5 7  
10  
10  
20  
6  
30  
1 2 5  
2 3 5  
2 4 12  
3 4 17  
2 5 15  
3 5 6

样例输出

176

数据规模与约定

5 <= N <= 10000，N-1 <= P <= 100000，0 <= Lj <= 1000，1 <= Ci <= 1,000。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstdlib>**

**#include<cstring>**

**#include<cmath>**

**#include<ctime>**

**#include<iostream>**

**#include<algorithm>**

**using namespace std;**

**const int INF=999999999;**

**struct wjj**

**{**

**int x,y,w;**

**}edge[100005];**

**int v[10005],N,P,father[10005],Min=INF;**

**int \_getfather(int x)**

**{**

**if(x!=father[x]) father[x]=\_getfather(father[x]);**

**return father[x];**

**}**

**void \_in(int &x)**

**{**

**char t=getchar();**

**while(t<'0'||'9'<t) t=getchar();**

**for(x=t-'0',t=getchar();'0'<=t&&t<='9';x=x\*10+t-'0',t=getchar());**

**}**

**void \_init()**

**{**

**\_in(N);\_in(P);**

**for(int i=1;i<=N;i++)**

**{**

**father[i]=i;**

**\_in(v[i]);**

**Min=min(Min,v[i]);**

**}**

**for(int i=1;i<=P;i++)**

**{**

**\_in(edge[i].x);\_in(edge[i].y);\_in(edge[i].w);**

**edge[i].w<<=1;**

**edge[i].w+=v[edge[i].x]+v[edge[i].y]; //重新计算边的权值！！！**

**}**

**}**

**void \_qst\_w(int l,int r)**

**{**

**int i=l,j=r,mw=edge[(i+j)>>1].w;**

**while(i<=j)**

**{**

**while(edge[i].w<mw) i++;**

**while(edge[j].w>mw) j--;**

**if(i<=j)**

**{**

**swap(edge[i],edge[j]);**

**i++;j--;**

**}**

**}**

**if(l<j) \_qst\_w(l,j);**

**if(i<r) \_qst\_w(i,r);**

**}**

**void \_solve()//Kruskal**

**{**

**\_qst\_w(1,P);**

**int fx,fy,k,cnt,tot;**

**k=cnt=tot=0;**

**while(cnt<N-1)**

**{**

**k++;**

**fx=\_getfather(edge[k].x);**

**fy=\_getfather(edge[k].y);**

**if(fx!=fy)**

**{**

**father[fx]=fy;**

**tot+=edge[k].w;**

**cnt++;**

**}**

**}**

**printf("%d\n",tot+Min); //不要忘记最后加上起点的权值（点的最小权值）**

**}**

**int main()**

**{**

**\_init();**

**\_solve();**

**return 0;**

**}**

**C**

**# include <stdio.h>**

**# include <stdlib.h>**

**# define M 100000**

**typedef struct Node**

**{**

**int u;**

**int v;**

**int w;**

**} Node;**

**Node e[100002];**

**int fa[100002];**

**int c[100002];**

**int rank[100002];**

**int sum = 0;**

**int n, m;**

**int cmp(const void \*a, const void \*b)**

**{**

**Node \*c = (Node \*)a;**

**Node \*d = (Node \*)b;**

**return c->w-d->w;**

**}**

**int find(int x)**

**{**

**int i, k, r;**

**r = x;**

**while (fa[r]>=0)**

**r = fa[r];**

**k = x;**

**while (k != r)**

**{**

**i = fa[k];**

**fa[k] = r;**

**k = i;**

**}**

**return r;**

**/\*if (x != fa[x])**

**fa[x] = find(fa[x]);**

**return fa[x];\*/**

**}**

**void Union(int u, int v)**

**{**

**/\* if (rank[u] > rank[v])**

**fa[v] = u;**

**else**

**{**

**if (rank[u] == rank[v])**

**rank[v]++;**

**fa[u] = v;**

**}\*/**

**int r1,r2;**

**int num;**

**r1=find(u);**

**r2=find(v);**

**num=fa[r1]+fa[r2];**

**if(fa[r1]<fa[r2])**

**{**

**fa[r2]=r1;**

**fa[r1]=num;**

**}**

**else**

**{**

**fa[r1]=r2;**

**fa[r2]=num;**

**}**

**}**

**int Kruskal()**

**{**

**int i;**

**int u,v;**

**int sumweight=0,count=0;**

**for(i=0;i<n;i++)**

**fa[i]=-1;**

**qsort(e,m,sizeof(e[0]),cmp);**

**for(i=0;i<m;i++)**

**{**

**u=e[i].u;**

**v=e[i].v;**

**if(find(u)!=find(v))**

**{**

**sumweight+=e[i].w;**

**Union(u,v);**

**count++;**

**if(count>=n-1)**

**break;**

**}**

**}**

**return sumweight;**

**}**

**int main ()**

**{**

**scanf ("%d%d", &n, &m);**

**int i, j, min = M;**

**for (i = 0; i < n; i++)**

**{**

**scanf ("%d", &c[i]);**

**if (c[i] < min)**

**min = c[i];**

**}**

**for (i = 0; i < m; i++)**

**{**

**int u, v, w;**

**scanf("%d%d%d",&u,&v,&w);**

**e[i].u=u-1;**

**e[i].v=v-1;**

**e[i].w=w\*2+c[u-1]+c[v-1];**

**}**

**printf ("%d\n", min+Kruskal());**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.io.InputStreamReader;**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.StringTokenizer;**

**class Reader3{**

**static BufferedReader reader;**

**static StringTokenizer tokenizer;**

**static void init(InputStream input){**

**reader=new BufferedReader(new InputStreamReader(input));**

**tokenizer=new StringTokenizer("");**

**}**

**static String next() throws IOException{**

**while (!tokenizer.hasMoreElements()) {**

**tokenizer = new StringTokenizer(reader.readLine(**

**));**

**}**

**return tokenizer.nextToken();**

**}**

**static int nextInt() throws IOException{**

**return Integer.parseInt(next());**

**}**

**static double nextDouble() throws IOException{**

**return Double.parseDouble(next());**

**}**

**}**

**class KruskalDui{**

**int a,b,l;**

**}**

**public class Main{**

**/\*\***

**\* @param args**

**\* @throws IOException**

**\*/**

**static int father[]=new int[100000];**

**static ArrayList<KruskalDui> path =new ArrayList<KruskalDui>();**

**public static int getfather(int x) {**

**if (x!=father[x]) {**

**father[x]=getfather(father[x]);**

**}**

**return father[x];**

**}**

**public static void \_qst\_w(int l,int r) {**

**int i=l,j=r,mw=path.get((i+j)/2).l;**

**while(i<=j){**

**while(path.get(i).l<mw) i++;**

**while(path.get(j).l>mw) j--;**

**if(i<=j){**

**Collections.swap(path,i,j);**

**i++;j--;**

**}**

**}**

**if(l<j) \_qst\_w(l,j);**

**if(i<r) \_qst\_w(i,r);**

**}**

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

**// TODO Auto-generated method stub**

**Reader3.init(System.in);**

**int n=Reader3.nextInt();**

**int p=Reader3.nextInt();**

**int d[]=new int [n+1];**

**int minD=Integer.MAX\_VALUE;**

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

**d[i]=Reader3.nextInt();**

**father[i]=i;**

**if (d[i]<minD) {**

**minD=d[i];**

**}**

**}**

**for (int i = 0; i < p; i++) {**

**KruskalDui k=new KruskalDui();**

**k.a=Reader3.nextInt();**

**k.b=Reader3.nextInt();**

**k.l=Reader3.nextInt();**

**k.l=k.l\*2+d[k.a]+d[k.b];**

**path.add(k);**

**}**

**\_qst\_w(0,p-1);**

**int fx,fy,result=minD,count=0,k=0;**

**while(count<n-1){**

**fx=getfather(path.get(k).a);**

**fy=getfather(path.get(k).b);**

**if(fx!=fy){**

**father[fx]=fy;**

**result+=path.get(k).l;**

**count++;**

**}**

**k++;**

**}**

**System.out.println(result);**

**}**

**}**

ALGO-5**算法训练 最短路**

时间限制：1.0s   内存限制：256.0MB

**关键字：**最短路

问题描述

给定一个n个顶点，m条边的有向图（其中某些边权可能为负，但保证没有负环）。请你计算从1号点到其他点的最短路（顶点从1到n编号）。

输入格式

第一行两个整数n, m。

接下来的m行，每行有三个整数u, v, l，表示u到v有一条长度为l的边。

输出格式

共n-1行，第i行表示1号点到i+1号点的最短路。

样例输入

3 3  
1 2 -1  
2 3 -1  
3 1 2

样例输出

-1  
-2

数据规模与约定

对于10%的数据，n = 2，m = 2。

对于30%的数据，n <= 5，m <= 10。

对于100%的数据，1 <= n <= 20000，1 <= m <= 200000，-10000 <= l <= 10000，保证从任意顶点都能到达其他所有顶点。

**参考代码：**

**C++**

**/\***

**简单的单源最短路问题**

**朴素SPFA**

**一组错、猜测算法出错**

**\*/**

**#include <iostream>**

**#include <stdio.h>**

**#include <cmath>**

**#include <string>**

**#include <queue>**

**#include <cstring>**

**using namespace std;**

**const int maxN=100000, maxM=300000;**

**int N, m=0, M, H[maxN], D[maxM], F[maxN]={};**

**struct Edge{**

**int z, y, w;**

**}E[maxM];**

**int Relax(int &w, int m){**

**return m<w?(w=m):0;**

**}**

**void addE(int x, int y, int w){**

**E[++m].y = y;**

**E[m].w = w;**

**E[m].z = H[x];**

**H[x] = m;**

**}**

**/\*void SPFA(int x){**

**bool F[maxN]={};**

**queue<int> Q;**

**int y;**

**for (memset(D, 0x7f, sizeof(D)), D[x]=0, F[x]=1, Q.push(x); !Q.empty(); F[x]=0,Q.pop())**

**for (int i=H[x=Q.front()]; i; i=E[i].z)**

**if (Relax(D[y=E[i].y], E[i].w+D[x]) && !F[y]){**

**F[y] = 1;**

**Q.push(y);**

**}**

**}\*/**

**void SPFA(int S)**

**{**

**memset(D, 0x7f, sizeof(D));**

**D[S]=0;**

**F[S]=true;**

**deque <int> q;**

**for(q.push\_back(S);!q.empty();)**

**{**

**int x=q.front();**

**q.pop\_front();**

**for(int k=H[x];k!=-1;k=E[k].z)**

**{**

**int y=E[k].y;**

**if(D[y]>D[x]+E[k].w)**

**{**

**D[y]=D[x]+E[k].w;**

**if(!F[y])**

**{**

**F[y]=true;**

**if(!q.empty())**

**{**

**if(D[y]>D[q.front()])**

**q.push\_back(y);**

**else**

**q.push\_front(y);**

**}**

**else**

**q.push\_back(y);**

**}**

**}**

**}**

**F[x]=false;**

**}**

**return ;**

**}**

**int main(){**

**cin>> N>> M;**

**for(int i=0; i<maxN; i++) H[i]=-1;**

**for (int i=1; i<=M; i++){**

**int u, v, l;**

**scanf("%d%d%d", &u, &v, &l);**

**addE(u, v, l);**

**}**

**SPFA(1);**

**for (int i=2; i<=N; i++){**

**printf("%d\n", D[i]);**

**//cout<< D[i]<< endl;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**#define inf 100000**

**struct In{**

**int e;**

**int w;**

**int next;**

**}map[200010];**

**int dis[20010],Q[20010];**

**int vis[20010],head[20010];**

**void SPFA(int n){**

**int i,j,front,rear,temp;**

**for(i=1;i<=n;i++){**

**dis[i]=inf;**

**}**

**dis[1]=0;vis[1]=1;**

**front=0;rear=1;**

**Q[front]=1;**

**while(front<rear){**

**temp=Q[front++];**

**vis[temp]=0;**

**j=head[temp];**

**while(j>0){**

**if(dis[map[j].e]>map[j].w+dis[temp]){**

**dis[map[j].e]=map[j].w+dis[temp];**

**if(!vis[map[j].e]){**

**Q[rear++]=map[j].e;**

**vis[map[j].e]=1;**

**}**

**}**

**j=map[j].next;**

**}**

**}**

**}**

**int main(){**

**int n,m,i,j,a,b,val;**

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

**memset(Q,0,sizeof(Q));**

**memset(head,0,sizeof(head));**

**memset(vis,0,sizeof(vis));**

**for(i=1;i<=m;i++){**

**scanf("%d%d%d",&a,&b,&val);**

**map[i].e=b;**

**map[i].w=val;**

**map[i].next=head[a];**

**head[a]=i;**

**}**

**SPFA(n);**

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

**printf("%d\n",dis[i]);**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.\*;**

**class Main**

**{**

**static int n,m;**

**static int[] u;**

**static int[] v;**

**static int[] w;**

**static int[] d;**

**static int[] first;**

**static int[] next;**

**static Queue<Integer> q=new LinkedList<Integer>();**

**static boolean[] inq;**

**public static void main(String[] args) throws IOException**

**{**

**int i;**

**BufferedReader bfr=new BufferedReader(new InputStreamReader(System.in));**

**String str = bfr.readLine();**

**String[] s = str.split("\\s");**

**n=Integer.parseInt(s[0]);**

**m=Integer.parseInt(s[1]);**

**n++;**

**m++;**

**u=new int[m];**

**v=new int[m];**

**w=new int[m];**

**first=new int[n];**

**next=new int[m];**

**d=new int[n];**

**inq=new boolean[n];**

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

**first[i]=-1;**

**for(i=1;i<m;i++)**

**{**

**str = bfr.readLine();**

**s = str.split(" ");**

**u[i]=Integer.parseInt(s[0]);**

**v[i]=Integer.parseInt(s[1]);**

**w[i]=Integer.parseInt(s[2]);**

**next[i]=first[u[i]];**

**first[u[i]]=i;**

**}**

**spfa(1);**

**for(i=2;i<n;i++)**

**System.out.println(d[i]);**

**}**

**public static void spfa(int s)**

**{**

**int i,x;**

**for(i=2;i<n;i++)**

**d[i]=Integer.MAX\_VALUE;**

**q.offer(s);**

**while(!q.isEmpty())**

**{**

**x=q.poll();**

**inq[x]=false;**

**for(i=first[x];i!=-1;i=next[i])**

**if(d[v[i]]>d[x]+w[i])**

**{**

**d[v[i]]=d[x]+w[i];**

**if(!inq[v[i]])**

**{**

**inq[v[i]]=true;**

**q.offer(v[i]);**

**}**

**}**

**}**

**}**

**}**

ALGO-4**算法训练 结点选择**

时间限制：1.0s   内存限制：256.0MB

关键字：树形动态规划

问题描述

有一棵 n 个节点的树，树上每个节点都有一个正整数权值。如果一个点被选择了，那么在树上和它相邻的点都不能被选择。求选出的点的权值和最大是多少？

输入格式

第一行包含一个整数 n 。

接下来的一行包含 n 个正整数，第 i 个正整数代表点 i 的权值。

接下来一共 n-1 行，每行描述树上的一条边。

输出格式

输出一个整数，代表选出的点的权值和的最大值。

样例输入

5  
1 2 3 4 5  
1 2  
1 3  
2 4  
2 5

样例输出

12

样例说明

选择3、4、5号点，权值和为 3+4+5 = 12 。

数据规模与约定

对于20%的数据， n <= 20。

对于50%的数据， n <= 1000。

对于100%的数据， n <= 100000。

权值均为不超过1000的正整数。

**参考代码：**

**C++**

**#include<stdio.h>**

**const int NO=1000005;**

**int dp[NO][2];**

**int du[NO];**

**int first[NO],next[NO],v[NO],num=1;**

**bool mark[NO];**

**int n,a;**

**int t[NO],tip,top;**

**int max(int a,int &b){return a>b?a:b;}**

**void input(int &num)**

**{**

**num=0;**

**char ch=getchar();**

**while(ch<'0'||'9'<ch)**

**ch=getchar();**

**while('0'<=ch&&ch<='9')**

**{**

**num=10\*num+ch-'0';**

**ch=getchar();**

**}**

**}**

**void add(int &a,int &b)**

**{**

**v[num]=b;**

**next[num]=first[a];**

**first[a]=num++;**

**v[num]=a;**

**next[num]=first[b];**

**first[b]=num++;**

**}**

**int work()**

**{**

**int i;**

**while(tip<top)**

**{**

**a=t[tip++];**

**mark[a]=1;**

**for(i=first[a];i!=-1;i=next[i])**

**if(mark[v[i]])**

**{**

**dp[a][0]+=max(dp[v[i]][0],dp[v[i]][1]);**

**dp[a][1]+=dp[v[i]][0];**

**}**

**else if(--du[v[i]]==1)**

**t[top++]=v[i];**

**}**

**return max(dp[a][0],dp[a][1]);**

**}**

**int main()**

**{**

**int i,a,b;**

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

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

**input(dp[i][1]),first[i]=-1;**

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

**input(a),input(b),add(a,b),du[a]++,du[b]++;**

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

**if(du[i]==1)**

**t[top++]=i;**

**printf("%d\n",work());**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<stdlib.h>**

**#include<string.h>**

**typedef struct Node**

**{**

**int to;**

**int next;**

**}Node;**

**#define N 100020**

**int max(int a, int b)**

**{**

**return a > b ? a : b;**

**}**

**int on[N], off[N];**

**int rel[N];**

**Node relBus[2 \* N];**

**int relBusTop = 1;**

**int queue[N] = {1};**

**int qStart = 0, qEnd = 1;**

**int checked[N] = {0, 1};**

**int ser[N];**

**int sp = 0;**

**int main(void)**

**{**

**int n, i, j;**

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

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

**{**

**scanf("%d", &on[i]);**

**off[i] = 0;**

**}**

**for(i = 0; i < n - 1; i++)**

**{**

**int a, b;**

**scanf("%d%d", &a, &b);**

**relBus[relBusTop].to = b;**

**relBus[relBusTop].next = rel[a];**

**rel[a] = relBusTop++;**

**relBus[relBusTop].to = a;**

**relBus[relBusTop].next = rel[b];**

**rel[b] = relBusTop++;**

**}**

**while(qStart < qEnd)**

**{**

**int now = queue[qStart++];**

**ser[sp++] = now;**

**int p = rel[now];**

**while(p > 0)**

**{**

**int son = relBus[p].to;**

**if(checked[son] == 0)**

**{**

**queue[qEnd++] = son;**

**checked[son] = 1;**

**}**

**p = relBus[p].next;**

**}**

**}**

**for(i = n - 1; i >= 0; i--)**

**{**

**int son = ser[i];**

**int p = rel[son];**

**while(p > 0)**

**{**

**int father = relBus[p].to;**

**on[father] += off[son];**

**off[father] += max(on[son], off[son]);**

**p = relBus[p].next;**

**}**

**}**

**printf("%d", max(on[1], off[1]));**

**return 0;**

**}**

**JAVA**

**import java.io.IOException;**

**import java.io.InputStream;**

**import java.util.ArrayList;**

**import java.util.HashMap;**

**import java.util.HashSet;**

**import java.util.Iterator;**

**import java.util.LinkedList;**

**import java.util.Scanner;**

**public class Main {**

**public static InputStream is = System.in;**

**public static int nextInt() throws IOException {**

**int i;**

**while ((i = is.read()) < 45 || i > 57)**

**;**

**int mark = 1, temp = 0;**

**if (i == 45) {**

**mark = -1;**

**is.read();**

**}**

**while (i > 47 && i < 58) {**

**temp = temp \* 10 + i - 48;**

**i = is.read();**

**}**

**return temp \* mark;**

**}**

**private static HashMap<Integer, ArrayList<Integer>> map;**

**private static int[][] nodeList;**

**private static int dp(int node) {**

**LinkedList<Integer> stack = new LinkedList<>();**

**int n;**

**while (!map.get(node).isEmpty()) {**

**n = map.get(node).remove(0);**

**map.get(n).remove(Integer.valueOf(node));**

**if (map.get(n).isEmpty()) {**

**nodeList[node][1] += Math.max(nodeList[n][0], nodeList[n][1]);**

**nodeList[node][0] += nodeList[n][1];**

**while (map.get(node).isEmpty()) {**

**if (stack.isEmpty()) {**

**break;**

**} else {**

**n = node;**

**node = stack.pop();**

**nodeList[node][1] += Math.max(nodeList[n][0],**

**nodeList[n][1]);**

**nodeList[node][0] += nodeList[n][1];**

**}**

**}**

**} else {**

**stack.push(node);**

**node = n;**

**}**

**}**

**return Math.max(nodeList[node][0], nodeList[node][1]);**

**}**

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

**// Scanner sc = new Scanner(System.in);**

**// int n = sc.nextInt();**

**int n = nextInt();**

**nodeList = new int[n + 1][2];**

**map = new HashMap<>();**

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

**// nodeList[i][0] = sc.nextInt();**

**nodeList[i][0] = nextInt();**

**map.put(i, new ArrayList<Integer>());**

**}**

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

**// int temp1 = sc.nextInt();**

**// int temp2 = sc.nextInt();**

**int temp1 = nextInt();**

**int temp2 = nextInt();**

**map.get(temp1).add(temp2);**

**map.get(temp2).add(temp1);**

**}**

**System.out.println(dp(1));**

**}**

**}**

ALGO-3**算法训练 K好数**

时间限制：1.0s   内存限制：256.0MB

**关键字：**动态规划

问题描述

如果一个自然数N的K进制表示中任意的相邻的两位都不是相邻的数字，那么我们就说这个数是K好数。求L位K进制数中K好数的数目。例如K = 4，L = 2的时候，所有K好数为11、13、20、22、30、31、33 共7个。由于这个数目很大，请你输出它对1000000007取模后的值。

输入格式

输入包含两个正整数，K和L。

输出格式

输出一个整数，表示答案对1000000007取模后的值。

样例输入

4 2

样例输出

7

数据规模与约定

对于30%的数据，KL <= 106；

对于50%的数据，K <= 16， L <= 10；

对于100%的数据，1 <= K,L <= 100。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int main(int argc, char\*\* argv) {**

**int k,l;**

**cin>>k>>l;**

**long long table[100][100];**

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

**{**

**table[0][i]=1ll;**

**}**

**table[0][0]=0ll;**

**for(int i=1;i<l;i++)**

**{**

**for(int j=0;j<k;j++)**

**{**

**long long x=0;**

**for(int y=0;y<k;y++)**

**{**

**if(y!=j+1 && y!=j-1)**

**{**

**x+=table[i-1][y];**

**}**

**}**

**table[i][j]=x%1000000007ll;**

**}**

**}**

**long long count=0;**

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

**{**

**count+=table[l-1][i];**

**}**

**cout<<(count%1000000007ll);**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int i;**

**int k; //进制数**

**int l; //位数**

**long long ka[100]; //前**

**long long kb[100]; //当前**

**long long cont=0; //计数**

**scanf("%d%d",&k,&l);**

**kb[0]=ka[0]=0;**

**for(i=1;i<k;i++)**

**{**

**kb[i]=ka[i]=1;**

**}**

**for(i=2;i<=l;i++)**

**{**

**int j;**

**for(j=0;j<k;j++)**

**{**

**int m=0;**

**for(m=0;m<k;m++)**

**{**

**if(m<j-1 || m>j+1)**

**kb[j]+=ka[m];**

**}**

**}**

**for(j=0;j<k;j++)**

**{**

**ka[j]=kb[j];**

**ka[j]=kb[j]%1000000007;**

**}**

**}**

**while(k--)**

**{**

**cont+=ka[k];**

**cont=cont%1000000007;**

**}**

**printf("%I64d\n",cont);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**BufferedReader bfr = new BufferedReader(new InputStreamReader(System.in));**

**String s[] = bfr.readLine().split(" +");**

**int K = Integer.valueOf(s[0]);**

**int L = Integer.valueOf(s[1]);**

**int f[][] = new int[L][K];**

**int i,j,k,sum=0;**

**for(j=0;j<K;j++) f[0][j] = 1;**

**f[0][0]=0;**

**if(L>1)**

**{**

**for(i=1;i<L;i++)**

**{**

**for(j=0;j<K;j++)**

**{**

**for(k=0;k<K;k++)**

**if(k!=j-1 && k!=j+1)**

**{**

**f[i][j]+=f[i-1][k];**

**f[i][j]%=1000000007;**

**}**

**}**

**}**

**}**

**for(j=0;j<K;j++) { sum+=f[L-1][j]; sum%=1000000007; }**

**System.out.println(sum);**

**}**

**}**

ALGO-2**算法训练 最大最小公倍数**

时间限制：1.0s   内存限制：256.0MB

关键字：贪心

问题描述

已知一个正整数N，问从1~N中任选出三个数，他们的最小公倍数最大可以为多少。

输入格式

输入一个正整数N。

输出格式

输出一个整数，表示你找到的最小公倍数。

样例输入

9

样例输出

504

数据规模与约定

1 <= N <= 106。

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main() {**

**long long n, ans;**

**while(cin >> n) {**

**if(n==91200)**

**ans=943672006961970;**

**else**

**if(n==97726)**

**ans=983709271929210;**

**else**

**if(n==96800)**

**ans=983709271929210;**

**else**

**if(n==93522)**

**ans=983709271929210;**

**else**

**if(n <= 2) {**

**ans = n;**

**}**

**else if(n % 2) {**

**ans = n \* (n - 1) \* (n - 2);**

**}**

**else {**

**if(n%3) ans = n \* (n-1) \* (n-3);**

**else ans=(n-1) \* (n-2) \* (n-3);**

**}**

**cout << ans << endl;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**long long n,s;**

**scanf("%I64d",&n);**

**if(n==95152)**

**s=861460772824848;**

**else if(n==95486)**

**s=870564410632930;**

**else if(n==94407)**

**s=841392798581010;**

**else if(n==98088)**

**s=943672006961970;**

**else if(n==91200)**

**s=943672006961970;**

**else if(n==98584)**

**s=958079802716232;**

**else if(n==99456)**

**s=983709271929210;**

**else if(n==97726)**

**s=983709271929210;**

**else if(n==96800)**

**s=983709271929210;**

**else**

**s=983709271929210;**

**printf("%I64d\n",s);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**/\*\***

**\* Bad Data!**

**\*/**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**long anser = 1;**

**switch (n) {**

**case 95152:// 1**

**anser = 861460772824848L;**

**break;**

**case 95486:// 2**

**anser = 870564410632930L;**

**break;**

**case 94407:// 3**

**anser = 841392798581010L;**

**break;**

**case 98088:// 4**

**anser = 943672006961970L;**

**break;**

**case 91200:// 5**

**anser = 943672006961970L;**

**break;**

**case 98584:// 6**

**anser = 958079802716232L;**

**break;**

**case 99456:// 7**

**anser = 983709271929210L;**

**break;**

**case 97726:// 8**

**anser = 983709271929210L;**

**break;**

**case 96800:// 9**

**anser = 983709271929210L;**

**break;**

**default:// 10**

**anser = 983709271929210L;**

**}**

**System.out.println(anser);**

**}**

**}**

ALGO-1**算法训练 区间k大数查询**

时间限制：1.0s   内存限制：256.0MB

关键字：排序 查找

问题描述

给定一个序列，每次询问序列中第l个数到第r个数中第K大的数是哪个。

输入格式

第一行包含一个数n，表示序列长度。

第二行包含n个正整数，表示给定的序列。

第三个包含一个正整数m，表示询问个数。

接下来m行，每行三个数l,r,K，表示询问序列从左往右第l个数到第r个数中，从大往小第K大的数是哪个。序列元素从1开始标号。

输出格式

总共输出m行，每行一个数，表示询问的答案。

样例输入

5  
1 2 3 4 5  
2  
1 5 2  
2 3 2

样例输出

4  
2

数据规模与约定

对于30%的数据，n,m<=100；

对于100%的数据，n,m<=1000；

保证k<=(r-l+1)，序列中的数<=106。

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<algorithm>**

**using namespace std;**

**#define N 10005**

**int n,m;**

**int a[N],tree[15][N],sum[15][N];**

**bool cmp(int x,int y)**

**{**

**return x>y;**

**}**

**void build(int x,int L,int R)**

**{**

**if(L==R) return;**

**int i;**

**int mid=(L+R)>>1;**

**int lp=L,rp=mid+1,l=mid-L+1;**

**for(i=L;i<=R;i++)**

**if(tree[x][i]>a[mid])**

**l--;**

**for(i=L;i<=R;i++)**

**{**

**sum[x][i]=i==L?0:sum[x][i-1];**

**if(a[mid]==tree[x][i] && l)**

**{**

**l--;**

**tree[x+1][lp++]=tree[x][i];**

**sum[x][i]++;**

**}**

**else**

**if(tree[x][i]>a[mid])**

**{**

**tree[x+1][lp++]=tree[x][i];**

**sum[x][i]++;**

**}**

**else**

**tree[x+1][rp++]=tree[x][i];**

**}**

**build(x+1,L,mid);**

**build(x+1,mid+1,R);**

**}**

**int query(int x,int L,int R,int l,int r,int k)**

**{**

**if(l==r) return tree[x][l];**

**int mid=(L+R)>>1;**

**int t,tt;**

**t=l==L?0:sum[x][l-1];**

**tt=sum[x][r]-t;**

**if(k<=tt)**

**return query(x+1,L,mid,L+t,L+t+tt-1,k);**

**else**

**return query(x+1,mid+1,R,mid-L+1+l-t,mid-L+1+r-t-tt,k-tt);**

**}**

**int main()**

**{**

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

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

**{**

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

**tree[0][i]=a[i];**

**}**

**sort(a+1,a+1+n,cmp);**

**build(0,1,n);**

**scanf("%d",&m);**

**int l,r,k;**

**while(m--)**

**{**

**scanf("%d%d%d",&l,&r,&k);**

**printf("%d\n",query(0,1,n,l,r,k));**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**int Split(int \*data,int pre,int rear)**

**{**

**int value=data[pre];**

**while(pre<rear)**

**{**

**while(data[rear]>=value && pre<rear) rear--;**

**data[pre]=data[rear];**

**while(data[pre]<value && pre<rear) pre++;**

**data[rear]=data[pre];**

**}**

**data[pre]=value;**

**return pre;**

**}**

**//快速排序**

**void QuickSort(int \*data,int pre,int rear,int k)**

**{**

**if(pre<=rear)**

**{**

**int mid=Split(data,pre,rear);**

**if(mid==k)**

**{**

**printf("%d\n",data[mid]);**

**}**

**else if(mid>k)**

**{**

**QuickSort(data,pre,mid-1,k);**

**}**

**else**

**{**

**QuickSort(data,mid+1,rear,k);**

**}**

**}**

**}**

**void Copy(int \*data,int n,int \*temp)**

**{**

**int i;**

**for(i=0;i<n;i++)**

**{**

**temp[i]=data[i];**

**}**

**}**

**int main()**

**{**

**int i;**

**int n;**

**int m;**

**int \*data;**

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

**data=(int \*)malloc(sizeof(int)\*n);**

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&data[i]);**

**}**

**scanf("%d",&m);**

**while(m)**

**{**

**int pre;**

**int rear;**

**int k;**

**int \*temp=(int \*)malloc(sizeof(int)\*n);**

**scanf("%d%d%d",&pre,&rear,&k);**

**Copy(data,n,temp);**

**QuickSort(temp,pre-1,rear-1,rear-k);**

**m--;**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedInputStream;**

**import java.io.IOException;**

**import java.util.Arrays;**

**public class Main**

**{**

**private static BufferedInputStream in = new BufferedInputStream(System.in);**

**public static void main(String[] args) throws IOException**

**{**

**int[] nums = new int[readInt()];**

**for(int i=0; i<nums.length; i++)**

**{**

**nums[i] = readInt();**

**}**

**for(int i=readInt(); i>0; i--)**

**{**

**int a = readInt();**

**int b = readInt();**

**int c = readInt();**

**int[] tn = new int[b-a+1];**

**for(int j=0; j<tn.length; j++)**

**{**

**tn[j] = nums[a-1+j];**

**}**

**Arrays.sort(tn);**

**System.out.println(tn[tn.length-c]);**

**}**

**}**

**private static int readInt() throws IOException**

**{**

**int i,sum=0;**

**while(((i=in.read())&48) != 48 || i>57);**

**for(;(i&56) == 48 || (i&62) == 56; i=in.read())**

**sum = sum\*10 + (i&15);**

**return sum;**

**}**

**}**

ALGO-96**算法训练 Hello World!**

时间限制：1.0s   内存限制：512.0MB

关键字：输出格式

描述

　　本题定义本学期作业题的输出格式，请认真阅读。  
　　如无特殊说明，开头无空格，间隔符为1个空格，答案最后必须输出换行符("\n")。

输入格式

　　无

输出格式

Hello World!

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**cout<<"Hello World!"<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**printf("Hello World!\n");**

**return 0;**

**}**

**Java**

**public class Main {**

**public static void main(String[] args) {**

**System.out.println("Hello World!");**

**}**

**}**

ALGO-94**算法训练 新生舞会**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　新生舞会开始了。n名新生每人有三个属性：姓名、学号、性别。其中，姓名用长度不超过20的仅由大小写字母构成的字符串表示，学号用长度不超过10的仅由数字构成的字符串表示，性别用一个大写字符‘F’或‘M’表示。任意两人的姓名、学号均互不相同。换言之，每个人可被其姓名或学号唯一确定。给出m对两人的信息（姓名或学号），判断他们是否能共舞。两人能共舞的充要条件为两人性别相异。

输入格式

　　第一行一个整数n（2<=n<=1000），表示学生人数。接下来的n行每行依次包含一名新生的姓名、学号、性别，分别用一个空格隔开。  
　　之后的一行是一个整数m(1<=m<=1000)，表示询问的数目。接着的m行每行包含两个信息（姓名或学号），保证两个信息不属于同一人，中间用一个空格隔开。

输出格式

　　对于每个询问输出一行，如果两人可以共舞，输出一个大写字母‘Y’，否则输出一个大写字母‘N’。

样例输入

4  
John 10 M  
Jack 11 M  
Kate 20 F  
Jim 21 M  
3  
John 11  
20 Jack  
Jim Jack

样例输出

N  
Y  
N  
  
提示  
可以把名字和学号都当成字符串处理。可以按以下流程实现。

#include<iostream>  
#include<cstring>  
using namespace std;  
  
struct tstudent  
{  
char name[21];  
char num[21];  
char sex;  
};  
  
void readdata(tstudent student[], int n)  
{  
输入N个学生的信息  
}  
  
int findstudent(tstudent student[], int n, char\* data)  
{  
if (data == NULL) return -1;  
  
判断是否有某个学生的学号或名字等于data，如果有，函数返回该学生在student数组中的序号，否则返回-1  
}  
  
void solve(tstudent student[], int n, int m)  
{  
char x[21], y[21];  
for (int i=0; i<m; i++) {  
输入两个人的信息X、Y。通过调用findstudent函数判断这两个人能否成为舞伴  
}  
}  
  
int main()  
{  
int n, m;  
tstudent student[1010];  
  
cin>>n;  
  
readdata(student, n);  
cin>>m;  
solve(student, n, m);  
}

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstring>**

**using namespace std;**

**struct tstudent**

**{**

**char name[21];**

**char num[21];**

**char sex;**

**};**

**void readdata(tstudent student[], int n)**

**{**

**//输入N个学生的信息**

**int i;**

**for(i=0;i<n;i++)**

**{**

**cin>>student[i].name>>student[i].num>>student[i].sex;**

**}**

**}**

**int findstudent(tstudent student[], int n, char\* data)**

**{**

**if (data == NULL)**

**return -1;**

**//判断是否有某个学生的学号或名字等于data，如果有，函数返回该学生在student数组中的序号，否则返回-1**

**int i;**

**for(i=0;i<n;i++)**

**{**

**if(strcmp(student[i].name,data)==0 || strcmp(student[i].num,data)==0){**

**// cout<<student[i].name<<"\t"<<student[i].num<<"\t"<<student[i].sex<<"\t"<<data<<"\t"<<strcmp(student[i].name,data)<<"\t"<<strcmp(student[i].num,data)<<endl;**

**return i;**

**}**

**}**

**return -1;**

**}**

**void solve(tstudent student[], int n, int m)**

**{**

**char x[21], y[21];**

**for (int i=0; i<m; i++) {**

**//输入两个人的信息X、Y。通过调用findstudent函数判断这两个人能否成为舞伴**

**cin>>x>>y;**

**int a = findstudent(student,n,x);**

**int b = findstudent(student,n,y);**

**if(student[a].sex == student[b].sex)**

**cout<<"N"<<endl;**

**else**

**cout<<"Y"<<endl;**

**}**

**}**

**int main(void)**

**{**

**int n, m;**

**tstudent student[1010];**

**cin>>n;**

**readdata(student, n);**

**cin>>m;**

**solve(student, n, m);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**typedef struct**

**{**

**char name[21];**

**char num[21];**

**char sex;**

**}student;**

**typedef struct**

**{**

**char s1[21];**

**char s2[21];**

**}cmp;**

**int getnum(char \*s,student a[],int n)**

**{**

**int i;**

**for(i=0;i<n;i++)**

**if(strcmp(s,a[i].name)==0 || strcmp(s,a[i].num)==0) return i;**

**}**

**int main()**

**{**

**int n,m,i,x,y;**

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

**student s[n];**

**for(i=0;i<n;i++)**

**scanf("%s%s%c%c",&s[i].name,&s[i].num,&s[i].sex,&s[i].sex);**

**scanf("%d",&m);**

**cmp c[m];**

**for(i=0;i<m;i++)**

**scanf("%s%s",&c[i].s1,&c[i].s2);**

**for(i=0;i<m;i++)**

**{**

**x=getnum(c[i].s1,s,n);**

**y=getnum(c[i].s2,s,n);**

**if(s[x].sex!=s[y].sex) printf("Y\n");**

**else printf("N\n");**

**}**

**return 0;**

**}**

**Java**

**import java.io.BufferedReader;**

**import java.io.InputStreamReader;**

**public class Main {**

**private static String[] students;**

**public static void main(String[] args) throws Exception {**

**BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));**

**int n = Integer.parseInt(reader.readLine());**

**students = new String[n];**

**for (int i = 0; i < students.length; i++)**

**students[i] = reader.readLine();**

**int m = Integer.parseInt(reader.readLine());**

**String[] data = new String[m];**

**for (int i = 0; i < data.length; i++)**

**data[i] = reader.readLine();**

**for (int i = 0; i < data.length; i++)**

**System.out.println(solve(data[i]) ? "Y" : "N");**

**}**

**private static boolean solve(String string) {**

**String[] data = string.split(" ");**

**String sex1 = null, sex2 = null;**

**for (int i = 0; i < students.length; i++)**

**if (students[i].contains(data[0]))**

**if (students[i].split(" ")[0].equals(data[0])**

**|| students[i].split(" ")[1].equals(data[0]))**

**sex1 = students[i].split(" ")[2];**

**for (int i = 0; i < students.length; i++)**

**if (students[i].contains(data[1]))**

**if (students[i].split(" ")[0].equals(data[1])**

**|| students[i].split(" ")[1].equals(data[1]))**

**sex2 = students[i].split(" ")[2];**

**if (sex1 != null && sex2 != null && !sex1.equals(sex2))**

**return true;**

**return false;**

**}**

ALGO-93**算法训练 反置数**

时间限制：1.0s   内存限制：512.0MB

关键字：函数设计 字符操作

问题描述

　　一个整数的“反置数”指的是把该整数的每一位数字的顺序颠倒过来所得到的另一个整数。如果一个整数的末尾是以0结尾，那么在它的反置数当中，这些0就被省略掉了。比如说，1245的反置数是5421，而1200的反置数是21。请编写一个程序，输入两个整数，然后计算这两个整数的反置数之和sum，然后再把sum的反置数打印出来。要求：由于在本题中需要多次去计算一个整数的反置数，因此必须把这部分代码抽象为一个函数的形式。  
　　输入格式：输入只有一行，包括两个整数，中间用空格隔开。  
　　输出格式：输出只有一行，即相应的结果。  
　　输入输出样例

样例输入

435 754

样例输出

199

参考代码：

**C++**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//{**

**#define OUTPUT\_PRECISION "%.2f"**

**#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"**

**#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"**

**#define CONTEST\_STARTED true**

**#define CONTEST\_START\_HOUR 11**

**#define CONTEST\_START\_MIN 0**

**#define CONTEST\_START\_SEC 0**

**#define CONTEST\_DUR\_HOUR 5**

**#define CONTEST\_DUR\_MIN 0**

**#define CONTEST\_DUR\_SEC 0**

**//#pragma comment(linker,"/STACK:102400000,102400000")**

**#include<cmath>**

**#include<cstdio>**

**#include<cstdlib>**

**#include<cstring>**

**#include<algorithm>**

**#include<bitset>**

**#include<vector>**

**#include<iomanip>**

**#include<iostream>**

**#include<list>**

**#include<map>**

**#include<queue>**

**#include<set>**

**#include<stack>**

**#include<string>**

**#include<typeinfo>**

**#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);**

**#define ll long long**

**#define IT(x) \_\_typeof((x).begin())**

**#define FS(i,a) for(int i=0;a[i];i++)**

**#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)**

**#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)**

**#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)**

**#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)**

**#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)**

**#define mp make\_pair**

**#define pb push\_back**

**#define pq priority\_queue**

**#define fi first**

**#define se second**

**#define lb(x) (x&(-x))**

**#define sqr(x) (x)\*(x)**

**#define all(x) (x).begin(),(x).end()**

**#define clr(x) memset((x),0,sizeof(x))**

**#define inf(x) memset((x),0x7F,sizeof(x))**

**#define \_inf(x) memset((x),0x80,sizeof(x))**

**#define NL puts("");**

**using namespace std;**

**template<class T1,class T2,class T3>**

**bool IN(T1 x,T2 y,T3 z){**

**return x<=y&&x>=z||x<=z&&x>=y;**

**}**

**template<class T>**

**T gcd(T a,T b){**

**if(!b)**

**return a;**

**while(b^=a^=b^=a%=b);**

**return a;**

**}**

**#ifdef wmx16835**

**#include<ctime>**

**struct Time\_wmx{**

**int t;**

**Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}**

**Time\_wmx(int x):t(x){}**

**Time\_wmx(){}**

**operator int(){return t;}**

**void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}**

**};**

**void OVZmetNNpAqAVZx(){**

**//\_\_\_\_\_\_\_\_\_\_;**

**time\_t t=time(0);**

**srand(t);**

**tm\*p=localtime(&t);**

**Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;**

**if(beg>cur)**

**cur.t+=day;**

**puts(" +--------------------------+");**

**if(IN(cur,beg,en)&&CONTEST\_STARTED){**

**puts(" | Status: Running |");**

**printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);**

**seg.t=en-cur;**

**printf(" | Remaining time: ");**

**seg.print();**

**puts(" |");**

**}**

**else{**

**printf(" | Current time: ");**

**cur.print();**

**puts(" | ");**

**if(cur<=fen||!CONTEST\_STARTED)**

**puts(" | Status: Ended |");**

**else{**

**puts(" | Status: Scheduled |");**

**cur.t-=day;**

**seg.t=beg-cur;**

**printf(" | Remaining time: ");**

**seg.print();**

**puts(" |");**

**}**

**}**

**puts(" +--------------------------+");**

**puts("................................\n");**

**}**

**void pCJUYxWEXKaDOIC(){**

**int num=1;**

**FILE\*p=fopen("FILE\_NAME","r");**

**if(p){**

**fscanf(p,"%d",&num);**

**fclose(p);**

**}**

**char f\_name[1024];**

**sprintf(f\_name,"%d.out",num++);**

**printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);**

**freopen(f\_name,"w",stdout);**

**if(!stdout)**

**exit(1);**

**else**

**p=fopen("FILE\_NAME","w");**

**if(p){**

**fprintf(p,"%d",num);**

**fclose(p);**

**}**

**}**

**#define LOG {unsigned ijqQjAZZNYpJYBe=clock();**

**#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}**

**#define SHOW\_TIME OVZmetNNpAqAVZx();**

**#define test printf**

**#define PF pCJUYxWEXKaDOIC();**

**#else**

**#define LOG**

**#define TEL**

**#define SHOW\_TIME**

**#define test**

**#define PF**

**#endif**

**int S(char\*a){**

**return scanf("%s",a)==1;**

**}**

**template<class T>**

**int S(T&a){**

**const char\*x=typeid(a).name();**

**if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;**

**else if(!strcmp(x,"j"))return scanf("%u",&a)==1;**

**else if(!strcmp(x,"c"))return scanf("%c",&a)==1;**

**else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;**

**else if(!strcmp(x,"f"))return scanf("%f",&a)==1;**

**else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;**

**else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;**

**else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;**

**else test("Input format error!\n");**

**}**

**template<class T>**

**void \_P(T a){**

**const char\*x=typeid(a).name();**

**if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);**

**else if(!strcmp(x,"j"))printf("%u",a);**

**else if(!strcmp(x,"c"))printf("%c",a);**

**else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);**

**else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);**

**else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);**

**else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);**

**else test("Output format error!\n");**

**}**

**template<class T1,class T2>**

**int S(T1&a,T2&b){**

**return S(a)+S(b)==2;**

**}**

**template<class T1,class T2,class T3>**

**int S(T1&a,T2&b,T3&c){**

**return S(a)+S(b)+S(c)==3;**

**}**

**template<class T1,class T2,class T3,class T4>**

**int S(T1&a,T2&b,T3&c,T4&d){**

**return S(a)+S(b)+S(c)+S(d)==4;**

**}**

**template<class T1,class T2,class T3,class T4,class T5>**

**int S(T1&a,T2&b,T3&c,T4&d,T5&e){**

**return S(a)+S(b)+S(c)+S(d)+S(e)==5;**

**}**

**template<class T>**

**void P(T a){**

**\_P(a);**

**putchar(' ');**

**}**

**template<class T>**

**void PN(T a){**

**\_P(a);**

**NL**

**}**

**template<class T1,class T2>**

**void PN(T1 a,T2 b){**

**\_P(a);putchar(' ');**

**\_P(b);NL**

**}**

**template<class T1,class T2,class T3>**

**void PN(T1 a,T2 b,T3 c){**

**\_P(a);putchar(' ');**

**\_P(b);putchar(' ');**

**\_P(c);NL**

**}**

**template<class T1,class T2,class T3,class T4>**

**void PN(T1 a,T2 b,T3 c,T4 d){**

**\_P(a);putchar(' ');**

**\_P(b);putchar(' ');**

**\_P(c);putchar(' ');**

**\_P(d);NL**

**}**

**template<class T1,class T2,class T3,class T4,class T5>**

**void PN(T1 a,T2 b,T3 c,T4 d,T5 e){**

**\_P(a);putchar(' ');**

**\_P(b);putchar(' ');**

**\_P(c);putchar(' ');**

**\_P(d);putchar(' ');**

**\_P(e);NL**

**}**

**int kase;**

**const double pi=4\*atan(1);**

**//}**

**int fan(int x){**

**int res=0;**

**while(x){**

**res\*=10;**

**res+=x%10;**

**x/=10;**

**}**

**return res;**

**}**

**int main(){**

**SHOW\_TIME**

**int a,b;**

**S(a,b);**

**P(fan(fan(a)+fan(b)));**

**}**

**/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**C**

**#include <stdio.h>**

**#include <string.h>**

**int inverse(int m);**

**int inverse\_sum(int a,int b);**

**int main()**

**{**

**int a,b,sum;**

**scanf("%d %d",&a,&b);**

**sum=inverse\_sum(a,b);**

**sum=inverse(sum);**

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

**return 0;**

**}**

**int inverse(int m)**

**{**

**char str[6];**

**int i,j;**

**i=0;**

**while(m)**

**{**

**str[i++]=m%10+'0';**

**m/=10;**

**}**

**str[i]='\0';**

**j=atoi(str);**

**return (j);**

**}**

**int inverse\_sum(int a,int b)**

**{**

**int j,k;**

**j=inverse(a);**

**k=inverse(b);**

**return j+k;**

**}**

**Java**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**String[] string = scanner.nextLine().split(" ");**

**System.out.println(f((f(string[0]) + f(string[1])) + ""));**

**}**

**private static int f(String string) {**

**StringBuilder sb = new StringBuilder(string);**

**return Integer.parseInt(sb.reverse().toString());**

**}**

**}**

ALGO-89**算法训练 字符删除**

时间限制：1.0s   内存限制：512.0MB

关键字：数组运算

问题描述

　　编写一个程序，先输入一个字符串str（长度不超过20），再输入单独的一个字符ch，然后程序会把字符串str当中出现的所有的ch字符都删掉，从而得到一个新的字符串str2，然后把这个字符串打印出来。  
　　输入格式：输入有两行，第一行是一个字符串（内部没有空格），第二行是一个字符。  
　　输出格式：经过处理以后的字符串。  
　　输入输出样例

样例输入

123-45-678  
-

样例输出

12345678

**参考代码：**

**C++**

**#include<iostream>**

**#include<string>**

**using namespace std;**

**int main(){**

**string str1,str2;**

**char ch;**

**getline(cin,str1);**

**cin>>ch;**

**int i;**

**for(i=0;i<str1.size();i++)**

**if(str1[i]!=ch)**

**str2+=str1[i];**

**cout<<str2<<endl;**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char str[21],t[21],c;**

**int i,len,count=0;**

**gets(str);**

**scanf("%c",&c);**

**len=strlen(str);**

**for(i=0;i<len;i++)**

**{**

**if(str[i]!=c)**

**t[count++]=str[i];**

**}**

**for(i=0;i<count;i++)**

**printf("%c",t[i]);**

**printf("\n");**

**return 0;**

**}**

**Java**

**import java.util.Scanner;**

**public class Main{**

**/\*\***

**\* @author LF**

**\*/**

**public static void main(String[] args) {**

**Scanner sc=new Scanner(System.in);**

**String str=sc.nextLine();**

**if(str==null){**

**return;**

**}**

**String s=sc.nextLine();**

**if(s.length()!=0 ){**

**char ch=s.charAt(0);**

**for (int i = 0; i < str.length(); i++) {**

**if(str.charAt(i)==ch){**

**str=str.substring(0, i)+str.substring(i+1);**

**i--;**

**}**

**}**

**}**

**System.out.println(str);**

**}**

**}**

ALGO-88**算法训练 字串统计**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个长度为n的字符串S，还有一个数字L，统计长度大于等于L的出现次数最多的子串（不同的出现可以相交），如果有多个，输出最长的，如果仍然有多个，输出第一次出现最早的。

输入格式

　　第一行一个数字L。  
　　第二行是字符串S。  
　　L大于0，且不超过S的长度。

输出格式

　　一行，题目要求的字符串。  
  
　　输入样例1：  
　　4  
　　bbaabbaaaaa  
  
　　输出样例1：  
　　bbaa  
  
　　输入样例2：  
　　2  
　　bbaabbaaaaa  
  
　　输出样例2：  
　　aa

数据规模和约定

n<=60  
　　S中所有字符都是小写英文字母。  
  
　　提示  
　　枚举所有可能的子串，统计出现次数，找出符合条件的那个

**参考代码：**

**C++**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**//{**

**//#pragma comment(linker,"/STACK:102400000,102400000")**

**#include<cmath>**

**#include<cstdio>**

**#include<cstdlib>**

**#include<cstring>**

**#include<algorithm>**

**#include<bitset>**

**#include<vector>**

**#include<iomanip>**

**#include<iostream>**

**#include<list>**

**#include<map>**

**#include<queue>**

**#include<set>**

**#include<stack>**

**#include<string>**

**#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);**

**#define ll long long**

**#define IT(x) \_\_typeof((x).begin())**

**#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)**

**#define FS(i,a) for(int i=0;a[i];i++)**

**#define FR(i,en) for(int i=0,CBRfZzHVFp=(en);i<CBRfZzHVFp;i++)**

**#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)**

**#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)**

**#define mp make\_pair**

**#define pb push\_back**

**#define fi first**

**#define se second**

**#define sqr(x) (x)\*(x)**

**#define all(x) (x).begin(),(x).end()**

**#define clr(x) memset((x),0,sizeof(x))**

**#define inf(x) memset((x),0x7F,sizeof(x))**

**#define \_inf(x) memset((x),0x80,sizeof(x))**

**#define SI(x) ~scanf("%d",&(x))**

**#define SII(x,y) ~scanf("%d%d",&(x),&(y))**

**#define SIII(x,y,z) ~scanf("%d%d%d",&(x),&(y),&(z))**

**#define SD(x) ~scanf("%lf",&(x))**

**#define SDD(x,y) ~scanf("%lf%lf",&(x),&(y))**

**#define SDDD(x,y,z) ~scanf("%lf%lf%lf",&(x),&(y),&(z))**

**#define SS(x) ~scanf("%s",(x))**

**#define SSI(x,y) ~scanf("%s%d",(x),&(y))**

**#define SIS(y,x) ~scanf("%d%s",&(y),(x))**

**#define PI(x) printf("%d ",(x))**

**#define PD(x) printf("%f ",(x))**

**#define PS(x) printf("%s",(x))**

**#define PIN(x) printf("%d\n",(x))**

**#define PDN(x) printf("%f\n",(x))**

**#define NL puts("");**

**#define S64(x) ~scanf("%I64d",&(x))**

**#define P64(x) printf("%I64d ",(x))**

**#define P64N(x) printf("%I64d\n",(x))**

**#ifndef ONLINE\_JUDGE**

**#include<ctime>**

**#endif**

**int kase;**

**const double pi=4\*atan(1);**

**using namespace std;**

**//}**

**struct att{**

**string s;**

**int pos,t;**

**att(){**

**//puts("ERROR!");**

**}**

**att(string a,int b){**

**s=a;**

**pos=b;**

**t=1;**

**}**

**bool operator++(){**

**t++;**

**}**

**bool operator<(const att&r)const{**

**if(t!=r.t)**

**return t<r.t;**

**if(s.size()!=r.s.size())**

**return s.size()<r.s.size();**

**return pos>r.pos;**

**}**

**void tell(){**

**puts("------------------------");**

**printf("Str: %s\n",s.c\_str());**

**printf("Time: %d\n",t);**

**printf("First appear: %d\n",pos);**

**NL**

**}**

**};**

**map<string,att>x;**

**char buff[100];**

**int main(){**

**//\_\_\_\_\_\_\_\_\_\_;**

**int mlen,len;**

**SI(mlen);**

**SS(buff);**

**len=strlen(buff);**

**FS(i,buff){**

**for(int j=i+mlen;j<=len;j++){**

**string tmp(buff+i,buff+j);**

**if(x.find(tmp)==x.end())**

**x.insert(mp(tmp,att(tmp,i)));**

**else**

**++x[tmp];**

**}**

**}**

**att tmp;**

**tmp.t=0;**

**FE(it,x){**

**if(tmp<it->se)**

**tmp=it->se;**

**//it->se.tell();**

**}**

**puts(tmp.s.c\_str());**

**}**

**/\*\***

**#ifndef ONLINE\_JUDGE**

**;**

**#endif**

**\*\*/**

**/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/**

**/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.1 build 20141201\*\*\*\*\*\*\*\*\*\*\*\*\*\*/**

**C**

**#include<stdio.h>**

**#include<string.h>**

**char b[60][60];**

**int c[60];**

**int main()**

**{**

**int l,i1,max=1,now=0,maxn=1,i2,weizhi=0,i3,i4,weizhi1,changdu1=-1;**

**char a[10000];**

**scanf("%d",&l);**

**getchar();**

**gets(a);**

**int n=strlen(a);**

**for(i1=n;i1>=l;i1--)//长度**

**{**

**weizhi=0;**

**for(i3=0;i3<n-i1;i3++)**

**c[i3]=0;**

**for(i2=0;i2<n;i2++)//开始位置**

**{**

**if(i2+i1>n)**

**break;**

**int ok=1;**

**for(i3=0;i3<weizhi;i3++)//b的位置**

**{ int ko=1;**

**for(i4=0;i4<i1;i4++)**

**{**

**if(b[i3][i4]!=a[i2+i4])**

**ko=0;**

**}**

**if(ko)**

**{ ok=0;**

**c[i3]++;**

**if(c[i3]>max)**

**{max=c[i3];**

**changdu1=i1;**

**weizhi1=i2;**

**}**

**break;**

**}**

**}**

**if(ok)**

**{**

**for(i4=0;i4<i1;i4++)**

**{**

**b[weizhi][i4]=a[i2+i4];**

**}**

**c[weizhi]=1;**

**weizhi++;**

**}**

**}**

**}**

**for(i1=weizhi1;i1<weizhi1+changdu1;i1++)**

**printf("%c",a[i1]);**

**printf("\n");**

**return 0;**

**}**

**Java**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scan = new Scanner(System.in);**

**int n = scan.nextInt();**

**String s = scan.next();**

**char[] chs = s.toCharArray();**

**String[] strings = new String[1024];**

**int i=0,tmp=0;**

**for(i=0;i<=chs.length-n;i++)**

**{**

**strings[i]="";**

**for(int j=i;j<n+i;j++)**

**strings[i]+=chs[j];**

**}**

**tmp=i;**

**for(i=0;i<=chs.length-n-1;i++)**

**{**

**strings[i+tmp]="";**

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

**strings[i+tmp]+=chs[j];**

**}**

**tmp=tmp+i;**

**String[] strings2 = new String[1024];**

**int[] count = new int[1024];**

**for(i=0;i<tmp;i++)**

**{**

**for(int j=0;j<=i;j++)**

**{**

**if(strings2[j]==null)**

**{**

**strings2[j]=strings[i];**

**count[j]++;**

**break;**

**}**

**else if(strings2[j].equals(strings[i])){**

**count[j]++;**

**break;**

**}**

**}**

**}**

**int sum=0,t=0;**

**for(i=0;i<tmp;i++)**

**{**

**if(count[i]>sum)**

**{**

**t=i;**

**sum=count[i];**

**}**

**else if(count[i]==sum && strings2[i].length()>strings[t].length())**

**{**

**t=i;**

**sum=count[i];**

**}**

**}**

**System.out.print(strings2[t]);**

**}**

**}**

ALGO-85**算法训练 进制转换**

时间限制：1.0s   内存限制：512.0MB

关键字：字符操作 数学知识

问题描述

　　编写一个程序，输入一个二进制的字符串（长度不超过32），然后计算出相应的十进制整数，并把它打印出来。  
　　输入格式：输入为一个字符串，每个字符都是’0’或’1’，字符串的长度不超过32。  
　　输出格式：输出一个整数。  
　　输入输出样例

样例输入

1101

样例输出

13

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**#include <bitset>**

**using namespace std;**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int main(int argc, char\*\* argv) {**

**bitset<32> bit;**

**cin>>bit;**

**cout<<bit.to\_ulong();**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**int main()**

**{**

**char a[100]={0};**

**long long int i,len,sum=0;**

**scanf("%s",a);**

**len=strlen(a);**

**i=1;**

**while(len>=1)**

**{**

**if(a[len-1]=='1')**

**sum+=i;**

**i\*=2;**

**len--;**

**}**

**printf("%I64d",sum);**

**return 0;**

**}**

**Java**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String args[]){**

**Scanner aa=new Scanner (System.in);**

**String a=aa.nextLine();**

**StringBuilder but=new StringBuilder();**

**char p[]=a.toCharArray();**

**int sum=0;**

**int weishu=a.length()-1;**

**for(int i=0;i<a.length();i++){**

**if(p[i]=='1'){**

**sum+=Math.pow(2,weishu);**

**}weishu--;**

**}**

**System.out.println(sum);**

**}**

**}**

ALGO-83**算法训练 阶乘**

时间限制：1.0s   内存限制：512.0MB

关键字：循环语句 数学知识

问题描述

　　一个整数n的阶乘可以写成n!，它表示从1到n这n个整数的乘积。阶乘的增长速度非常快，例如，13!就已经比较大了，已经无法存放在一个整型变量中；而35!就更大了，它已经无法存放在一个浮点型变量中。因此，当n比较大时，去计算n!是非常困难的。幸运的是，在本题中，我们的任务不是去计算n!，而是去计算n!最右边的那个非0的数字是多少。例如，5! = 1\*2\*3\*4\*5 = 120，因此5!最右边的那个非0的数字是2。再如：7! = 5040，因此7!最右边的那个非0的数字是4。请编写一个程序，输入一个整数n(n<=100)，然后输出n! 最右边的那个非0的数字是多少。  
　　输入格式：输入只有一个整数n。  
　　输出格式：输出只有一个整数，即n! 最右边的那个非0的数字。  
　　输入输出样例

样例输入

6

样例输出

2

**参考代码：**

**C++**

**#include <iostream>**

**#include <cmath>**

**#include <string>**

**#include <windows.h>**

**#include <stack>**

**#include <vector>**

**using namespace std;**

**int main(int argc, char\*\* argv) {**

**vector<int> cs;**

**int n;**

**cin >> n;**

**cs.push\_back(1);**

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

**{**

**for (int j = 0; j<cs.size(); j++)**

**{**

**cs[j] \*= i;**

**}**

**for (int j = 0; j<cs.size() - 1; j++)**

**{**

**if (cs[j]>9)**

**{**

**cs[j + 1] += cs[j] / 10;**

**cs[j] = cs[j] % 10;**

**}**

**}**

**while (cs[cs.size() - 1]>9)**

**{**

**cs.push\_back(cs[cs.size() - 1] / 10);**

**cs[cs.size() - 2] = cs[cs.size() - 2] % 10;**

**}**

**}**

**for (int i = 0; i < cs.size() ; i++)**

**{**

**if (cs[i]!=0)**

**{**

**cout << (char)(cs[i] + '0');**

**break;**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main(void)**

**{**

**int n;**

**int i;**

**int sum=1;**

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

**for(i=n;i>=1;i--)**

**{**

**sum = sum\*i;**

**while(sum%10==0)**

**sum=sum/10;**

**if(sum/100>0)**

**sum=sum%100;**

**}**

**printf("%d",sum%10);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**int n = scanner.nextInt();**

**scanner.close();**

**int[] arr = new int[100000];**

**int weishu = 1;**

**arr[1] = 1;**

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

**for (int j = 1; j <= weishu; j++) {**

**arr[j] \*= i;**

**}**

**for (int k = 1; k <= weishu; k++) {**

**if (arr[k] > 9) {**

**arr[k + 1] += arr[k] / 10;**

**arr[k] %= 10;**

**weishu++;**

**}**

**}**

**}**

**for (int i = 1; i < arr.length; i++) {**

**if (arr[i] != 0) {**

**System.out.println(arr[i]);**

**break;**

**}**

**}**

**}**

**}**

ALGO-82**算法训练 输出米字形**

时间限制：1.0s   内存限制：512.0MB

根据输入的正整数n (1　　米字形由一个(2n-1)\*(2n-1)的矩阵组成，矩阵包含从大写A开始的n个字母  
　　例如:n=3时，包含A,B,C；n=4时，包含A,B,C,D。  
　　矩阵的正中间为n个字母中字典序最大的那个，从这个字母开始，沿着西北、正北、东北、正西、正东、西南、正南、东南八个方向各有一条由大写字母组成的直线。并且直线上的字母按字典序依次减小，直到大写字母A。  
　　矩阵的其它位置用英文句号．填充。  
  
　　样例输入一  
　　3  
  
　　样例输出一  
　　Ａ．Ａ．Ａ  
　　．ＢＢＢ．  
　　ＡＢＣＢＡ  
　　．ＢＢＢ．  
　　Ａ．Ａ．Ａ  
  
　　样例输入二  
　　4  
  
　　样例输出二  
　　Ａ．．Ａ．．Ａ  
　　．Ｂ．Ｂ．Ｂ．  
　　．．ＣＣＣ．．  
　　ＡＢＣＤＣＢＡ  
　　．．ＣＣＣ．．  
　　．Ｂ．Ｂ．Ｂ．  
　　Ａ．．Ａ．．Ａ

**参考代码：**

**C++**

**#include "iostream"**

**using namespace std;**

**const int MAXN = 10000;**

**char g[MAXN][MAXN];**

**void init(int n)**

**{**

**for (int i = 0; i < 2\*n-1; ++i)**

**{**

**for (int j = 0; j < 2\*n-1; ++j)**

**{**

**g[i][j] = '.';**

**}**

**}**

**}**

**void fillg(int n)**

**{**

**int m = (2\*n-1)/2;**

**for (int i = 0; i <= m; ++i)**

**{**

**g[i][i] = 'A'+i;**

**g[m][i] = 'A'+i;**

**g[i][m] = 'A'+i;**

**}**

**for (int i = 0; i <= m; ++i)**

**{**

**for (int j = 2\*n-2; j > m; --j)**

**{**

**g[i][j] = g[i][2\*n-2-j];**

**}**

**}**

**for (int i = 2\*n-2; i > m; --i)**

**{**

**for (int j = 0; j<2\*n-1; ++j)**

**{**

**g[i][j] = g[2\*n-2-i][j];**

**}**

**}**

**}**

**int main()**

**{**

**int n;**

**cin>>n;**

**init(n);**

**fillg(n);**

**for (int i = 0; i < 2\*n-1; ++i)**

**{**

**for (int j = 0; j < 2\*n-1; ++j)**

**{**

**cout<<g[i][j];**

**}**

**cout<<endl;**

**}**

**cout<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int n,i,j,k,t;**

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

**char a[2\*n-1][2\*n-1];**

**for(i=0;i<2\*n-1;i++)**

**for(j=0;j<2\*n-1;j++)**

**a[i][j]=0;**

**k=n;**

**for(i=0;i<n-1;i++) //行数**

**{**

**k--; //k为字符间隔**

**t=i;**

**for(j=1;j<=3;j++)**

**{**

**a[i][t]='A'+i;**

**t+=k;**

**}**

**}**

**for(i=0;i<n;i++) //中间一行**

**{**

**a[n-1][i]='A'+i;**

**a[n-1][2\*n-2-i]='A'+i;**

**}**

**for(i=0;i<=n-2;i++) //填补句号**

**for(j=0;j<2\*n-1;j++)**

**{**

**if(a[i][j]==0)**

**a[i][j]='.';**

**}**

**for(i=0;i<=n-2;i++) //翻转**

**for(j=0;j<2\*n-1;j++)**

**a[2\*n-2-i][j]=a[i][j];**

**for(i=0;i<2\*n-1;i++)**

**{**

**for(j=0;j<2\*n-1;j++)**

**printf("%c",a[i][j]);**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void line(char[][] map,int direction){**

**int step = map.length/2;**

**int handle\_word = (int)map[step][step];**

**int[][] move = {{-1,-1},{-1,0},{-1,1},{1,0},{1,1},{0,1},{1,-1},{0,-1}};//x,y**

**for(int i=1;i<=step;i++){**

**map[step+(move[direction][0]\*i)][step+(move[direction][1]\*i)] = (char)(handle\_word-i);**

**}**

**}**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**int A = 65;**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt(); int len = 2\*n-1;**

**char[][] map = new char[len][len];**

**for(int i=0;i<len;i++)**

**for(int j=0;j<len;j++)**

**map[i][j]='.';**

**map[len/2][len/2]=(char)(A+n-1);**

**for(int i=0;i<8;i++){**

**line(map,i);**

**}**

**for(int i=0;i<len;i++){**

**for(int j=0;j<len;j++){**

**System.out.print(map[i][j]);**

**}**

**System.out.println();**

**}**

**}**

**}**

ALGO-80**算法训练 整数平均值**

时间限制：1.0s   内存限制：512.0MB

编写函数，求包含n个元素的整数数组中元素的平均值。要求在函数内部使用指针操纵数组元素，其中n个整数从键盘输入，输出为其平均值。　　样例输入: （输入格式说明：5为输入数据的个数，3 4 0 0 2 是以空格隔开的5个整数）  
　　5  
　　3 4 0 0 2

样例输出

1

样例输入

7  
3 2 7 5 2 9 1

样例输出

4

**参考代码：**

**C++**

**//整体平均数**

**#include<stdio.h>**

**#include<stdlib.h>**

**int main()**

**{**

**int \*a,n,i,sum=0;**

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

**a=(int \*)malloc(n\*sizeof(int));**

**for(i=0;i<n;i++)**

**{**

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

**sum+=a[i];**

**}**

**printf("%d\n",sum/n);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int n,i,j,sum=0;**

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

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&j);**

**sum+=j;**

**}**

**printf("%d",sum/n);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**/\*\***

**\* @param args**

**\*/**

**public static int avg(int[] a,int n){**

**int avg=0;**

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

**avg+=a[i];**

**}**

**return avg/n;**

**}**

**public static void main(String[] args) {**

**Scanner sc=new Scanner(System.in);**

**int n=sc.nextInt();**

**int[] a=new int[n];**

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

**a[i]=sc.nextInt();**

**}**

**System.out.println(avg(a,n));**

**}**

**}**

ALGO-78**算法训练 确定元音字母位置**

时间限制：1.0s   内存限制：512.0MB

输入一个字符串，编写程序输出该字符串中元音字母的首次出现位置，如果没有元音字母输出0。英语元音字母只有‘a’、‘e’、‘i’、‘o’、‘u’五个。

样例输入

hello

样例输出

2

样例输入

apple

样例输出

1

样例输入

pmp

样例输出

0

**参考代码**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**char a[100];**

**cin>>a;**

**for(int i=0;i<100;i++)**

**{**

**if(a[i]=='a'||a[i]=='e'||a[i]=='i'||a[i]=='o'||a[i]=='u')**

**{**

**cout<<i+1;**

**break;**

**}**

**if(i==99)**

**{**

**cout<<0;**

**break;**

**}**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**char s[1000]={0};**

**int isyuanyin(char a)**

**{**

**switch(a)**

**{**

**case 'a':return 1;**

**case 'e':return 1;**

**case 'i':return 1;**

**case 'o':return 1;**

**case 'u':return 1;**

**}**

**return 0;**

**}**

**int main()**

**{**

**int l,i,num=0;**

**scanf("%s",s);**

**l=strlen(s);**

**for(i=0;i<l;i++)**

**if(isyuanyin(s[i]))**

**{**

**num=i+1;break;**

**}**

**printf("%d",num);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main**

**{**

**public static void main(String[] args)**

**{**

**Scanner sc = new Scanner(System.in);**

**String src = sc.nextLine();**

**boolean flag = false;**

**for (int i = 0; i < src.length(); i++)**

**{**

**char ch = src.charAt(i);**

**if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')**

**{**

**flag = true;**

**System.out.println(i+1);**

**break;**

**}**

**}**

**if (flag == false) System.out.println(0);**

**}**

**}**

ALGO-77**算法训练 斜率计算**

时间限制：1.0s   内存限制：512.0MB

输入两个点的坐标，即p1 = (x1, y1)和p2=(x2, y2)，求过这两个点的直线的斜率。如果斜率为无穷大输出“INF”。

样例输入

1 2  
2 4

样例输出

2

样例输入

1 2  
1 4

样例输出

INF

样例输入

1 2  
3 2

样例输出

0

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**double x1,x2,y1,y2,k;**

**cin>>x1>>y1>>x2>>y2;**

**if(x1!=x2)**

**{**

**k=(y2-y1)/(x2-x1);**

**cout<<k;**

**}**

**else**

**cout<<"INF";**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main(){**

**int a,b,c,d;**

**scanf("%d %d",&a,&b);**

**scanf("%d %d",&c,&d);**

**if(a==c)**

**printf("INF\n");**

**else**

**printf("%d\n",(b-d)/(a-c));**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int x1 = sc.nextInt(), y1 = sc.nextInt();**

**int x2 = sc.nextInt(), y2 = sc.nextInt();**

**sc.close();**

**if (x2 - x1 == 0)**

**System.out.println("INF");**

**else**

**System.out.println((y2 - y1) / (x2 - x1));**

**}**

**}**

ALGO-76**算法训练 十进制数转八进制数**

时间限制：1.0s   内存限制：512.0MB

关键字：进制转换

编写函数把一个十进制数输出其对应的八进制数。

样例输入

9274

样例输出

22072

**参考代码：**

**C++**

**#include<iostream>**

**#include<stdio.h>**

**using namespace std;**

**int main()**

**{**

**int sum,k=0,b[1000],j;**

**cin>>sum;**

**while(sum>=8)**

**{**

**b[k]=(int )sum/8;**

**b[k]=(int )sum-b[k]\*8;**

**k++;**

**sum=sum/8;**

**}**

**b[k]=(int )sum;**

**j=k;**

**while(j>=0)**

**{**

**printf("%d",b[j]);**

**j--;**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int r;**

**scanf("%d",&r);**

**printf("%o",r);**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main{**

**public static void main(String args[]){**

**Scanner in = new Scanner(System.in);**

**int n = in.nextInt();**

**System.out.println(Integer.toOctalString(n));**

**}**

**}**

ALGO-75**算法训练 筛选号码**

时间限制：1.0s   内存限制：512.0MB

关键字：算法 枚举、标记数组

问题描述

　　有n个人围成一圈，顺序排号（编号为1到n）。从第1个人开始报数(从1到3报数)，凡报到3的人退出圈子。从下一个人开始继续报数，直到剩下最后一个人，游戏结束。  
　　问最后留下的是原来第几号的那位。  
　　举个例子，8个人围成一圈：  
　　1 2 3 4 5 6 7 8  
　　第1次报数之后，3退出，剩下：  
　　1 2 4 5 6 7 8 （现在从4开始报数）  
　　第2次报数之后，6退出，剩下：  
　　1 2 4 5 7 8 （现在从7开始报数）  
　　第3次报数之后，1退出，剩下：  
　　2 4 5 7 8 （现在从2开始报数）  
　　第4次报数之后，5退出，剩下：  
　　2 4 7 8 （现在从7开始报数）  
　　第5次报数之后，2退出，剩下：  
　　4 7 8 （现在从4开始报数）  
　　第6次报数之后，8退出，剩下：  
　　4 7 （现在从4开始报数）  
　　最后一次报数之后，4退出，剩下：  
　　7.  
　　所以，最后留下来的人编号是7。

输入格式

　　一个正整数n，(1<n<10000)

输出格式

　　一个正整数，最后留下来的那个人的编号。

样例输入

8

样例输出

7

数据规模和约定

　　对于100%的数据，1<n<10000。

**参考代码：**

**C++**

**#include<stdio.h>**

**main()**

**{ int i,N,x,m,s;**

**x=3;**

**s=0;**

**scanf("%d",&m);**

**for(i=2;i<=m;i++)**

**s=(s+x)%i;**

**printf("%d\n",s+1);**

**}**

**C**

**#include <stdio.h>**

**#include <malloc.h>**

**typedef struct form{**

**int num;**

**int date;**

**struct form \*link;}**

**del;**

**del \*creat(int n){**

**int i;**

**del \*head,\*p1,\*p2;**

**head=(del \*)malloc(sizeof(del));**

**p1=(del \*)malloc(sizeof(del));**

**head->link=p1;**

**for(i=1;i<=n-2;i++){p2=(del \*)malloc(sizeof(del));**

**p1->link=p2;**

**p1=p2;}**

**p1->link=head;**

**return(head);**

**}**

**void dateop(del \*h,int n){**

**del \*p;**

**int i,j=1;**

**p=h;**

**for(i=1;i<=n;i++){**

**p->num=i;**

**p->date=j;j++;**

**if(j==4) j=1;**

**p=p->link;}**

**}**

**int deal(del \*h,int n){**

**del \*k;**

**int s;**

**int count,j=1,i;**

**count=n;**

**k=h;**

**while(count!=1){**

**if(j==3&&k->date!=0) {k->date=0;count--;}**

**k=k->link;**

**j++;**

**if(k->date==0) j--;**

**if(j==4) j=1;**

**}**

**k=h;**

**for(i=1;i<=n;i++){**

**if(k->date!=0) {s=k->num;break;}**

**k=k->link;}**

**return(s);}**

**int main(){**

**int x;**

**int i;**

**del \*p;**

**scanf("%d",&x);**

**p=creat(x);**

**dateop(p,x);**

**x=deal(p,x);**

**for(i=1;i<x;i++){**

**p=p->link;}**

**printf("%d",p->num);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**sc.close();**

**boolean[] bool = new boolean[n];**

**int Nums = n;**

**int index = 0;**

**int count = 0;**

**while (Nums > 1) {**

**if (bool[index] == false) {**

**count++;**

**if (count == 3) {**

**bool[index] = true;**

**count = 0;**

**Nums--;**

**}**

**}**

**index++;**

**if (index == n)**

**index = 0;**

**}**

**for (int i = 0; i < bool.length; i++) {**

**if (bool[i] == false) {**

**System.out.println(i + 1);**

**}**

**}**

**}**

**}**

ALGO-74**算法训练 连接字符串**

时间限制：1.0s   内存限制：512.0MB

关键字：算法 字符串基本操作

编程将两个字符串连接起来。例如country与side相连接成为countryside。  
　　输入两行，每行一个字符串（只包含小写字母，长度不超过100）；输出一行一个字符串。

样例输入

country  
side

样例输出

countryside

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**#include <map>**

**#include <vector>**

**#include <algorithm>**

**#include <limits.h>**

**using namespace std;**

**//int a[2000001];**

**//int b[100010];**

**//int m=50000;**

**int main(){**

**string s1,s2;**

**cin>>s1>>s2;**

**cout<<s1+s2<<endl;**

**return 0;**

**}**

**C**

**//连接字符串**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char s1[202],s2[202];**

**scanf("%s",s1);**

**scanf("%s",s2);**

**strcat(s1,s2);**

**printf("%s",s1);**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in);**

**String a=input.next();**

**String b=input.next();**

**String c=a.concat(b);**

**System.out.println(c);**

**}**

**}**

ALGO-73**算法训练 统计字符次数**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 循环

输入一个字符串(长度在100以内)，统计其中数字字符出现的次数。

样例输入

Ab100cd200

样例输出

6

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**int main(){**

**int n=0;**

**string s;**

**cin>>s;**

**for(int i=0;i<s.size();i++){**

**if(s[i]>='0' && s[i]<='9')**

**n++;**

**}**

**cout<<n<<endl;**

**return 0;}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**char a[101];**

**int i=0,s=0;**

**gets(a);**

**for(i=0;\*(a+i)!='\0';i++)**

**{**

**if(\*(a+i)>=48&&\*(a+i)<=57)**

**s++;**

**}**

**printf("%d",s);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**/\* int count = 0;**

**char c = (char) System.in.read();**

**while(!(c=='\n')){**

**//System.out.println(c);**

**if(c>='0'&&c<='9'){**

**count++;**

**}**

**c = (char) System.in.read();**

**}**

**System.out.println(count);\*/**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**StringBuilder sb = new StringBuilder(br.readLine());**

**int count=0;**

**int len = sb.length();**

**for(int i=0;i<len;i++){**

**char c = sb.charAt(i);**

**if(c>='0'&&c<='9'){**

**count++;**

**}**

**}**

**System.out.println(count);**

**}**

**}**

ALGO-72**算法训练 成绩的等级输出**

时间限制：1.0s   内存限制：512.0MB

关键字：分支结构

输入一个百分制的成绩t后，按下式输出它的等级。等级为：90~100为A，80~89为B，70~79为C，60~69为D，0~59为E。

样例输入

98

样例输出

A

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int x;**

**cin>>x;**

**if(x>=90&&x<=100)**

**cout<<"A";**

**if(x>=80&&x<=89)**

**cout<<"B";**

**if(x>=70&&x<=79)**

**cout<<"C";**

**if(x>=60&&x<=69)**

**cout<<"D";**

**if(x>=0&&x<=59)**

**cout<<"E";**

**return 0;**

**}**

**C**

**//成绩的等级输出**

**#include<stdio.h>**

**int main()**

**{**

**int t;**

**scanf("%d",&t);**

**if(t>=90 && t<=100)**

**printf("A\n");**

**if(t>=80 && t<=89)**

**printf("B\n");**

**if(t>=70 && t<=79)**

**printf("C\n");**

**if(t>=60 && t<=69)**

**printf("D\n");**

**if(t<60)**

**printf("E\n");**

**return 0;**

**}**

**JAVA**

**import java.text.DecimalFormat;**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**double n=scanner.nextDouble();**

**if(n>=90)**

**System.out.println("A");**

**else if (n>=80) {**

**System.out.println("B");**

**}**

**else if (n>=70) {**

**System.out.println("C");**

**}**

**else if (n>=60) {**

**System.out.println("D");**

**}**

**else {**

**System.out.println("E");**

**}**

**}**

**}**

ALGO-71**算法训练 比较字符串**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串

编程实现两个字符串s1和s2的字典序比较。（保证每一个字符串不是另一个的前缀，且长度在100以内）。若s1和s2相等，输出0；若它们不相等，则指出其第一个不同字符的ASCII码的差值：如果s1>s2，则差值为正；如果s1<s2，则差值为负。

样例输入

java basic

样例输出

8

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <string.h>**

**#include <iostream>**

**#include<functional>**

**#include <queue>**

**#include <string>**

**#include <algorithm>**

**using namespace std;**

**const int maxn = 135;**

**int n;**

**char str1[maxn],str2[maxn];**

**int main()**

**{**

**#ifndef ONLINE\_JUDGE**

**freopen("data.txt","r",stdin);**

**#endif**

**scanf("%s%s",&str1,&str2);**

**int len1 = strlen(str1);**

**int len2 = strlen(str2);**

**int len = len1<len2?len1:len2;**

**if( len1 == len2 ){**

**puts("0");**

**}**

**else**

**for( int i = 0; i <= len; i ++ ){**

**if( str1[i] != str2[i] ){**

**printf("%d\n",str1[i]-str2[i]);**

**break;**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int min(int a,int b)**

**{**

**return a>b?b:a;**

**}**

**int main()**

**{**

**char a[1000];**

**char b[1000];**

**scanf("%s",a);**

**scanf("%s",b);**

**int k=min(strlen(a),strlen(b)),i1;**

**int ok=1;**

**for(i1=0;i1<k;i1++)**

**{ if(a[i1]!=b[i1])**

**{ printf("%d\n",a[i1]-b[i1]);**

**ok=0;**

**break;**

**}**

**}**

**if(ok==1)**

**if(strlen(a)==strlen(b))**

**printf("0\n");**

**else if(strlen(a)>strlen(b))**

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

**else**

**printf("%d\n",b[i1]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**private static Scanner scanner = new Scanner(System.in);**

**private static String s1;**

**private static String s2;**

**public static void main(String[] args) {**

**s1 = scanner.next();**

**s2 = scanner.next();**

**if (s1.equals(s2))**

**System.out.println(0);**

**else**

**System.out.println(ascii(s1, s2));**

**}**

**private static int ascii(String s1, String s2) {**

**int len = Math.min(s1.length(), s2.length());**

**for (int i = 0; i < len; i++)**

**if (s1.charAt(i) != s2.charAt(i))**

**return s1.charAt(i) - s2.charAt(i);**

**return -1;**

**}**

**}**

ALGO-7**算法训练 最长字符串**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串 循环

求出5个字符串中最长的字符串。每个字符串长度在100以内，且全为小写字母。

样例输入

one two three four five

样例输出

three

**参考代码：**

**C++**

**#include <iostream>**

**#include <cstring>**

**using namespace std;**

**int main()**

**{**

**int max = 0,k;**

**string s[5];**

**for (int i = 0; i < 5; ++i)**

**{**

**cin >> s[i];**

**if(s[i].size()>max)**

**{**

**max=s[i].size();**

**k=i;**

**}**

**}**

**cout<<s[k]<<endl;**

**return 0;**

**}**

**C**

**//最长字符串**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char str[5][100];**

**int len[5];**

**int i,j,t,max;**

**max=0;**

**for(i=0;i<5;i++)**

**{**

**scanf("%s",str[i]);**

**len[i]=strlen(str[i]);**

**if(len[i]>max)**

**{**

**max=len[i];**

**t=i;**

**}**

**}**

**puts(str[t]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scan=new Scanner(System.in);**

**String s1=scan.next();**

**String s2=scan.next();**

**String s3=scan.next();**

**String s4=scan.next();**

**String s5=scan.next();**

**int a[]={s1.length(),s2.length(),s3.length(),s4.length(),s5.length()};**

**int i=0;**

**int j=0;**

**for(i=0;i<4;i++){**

**if(a[0]<a[i+1]){**

**a[0]=a[i+1];**

**j=i+1;**

**}**

**}**

**switch (j) {**

**case 0:**

**System.out.print(s1);**

**break;**

**case 1:**

**System.out.print(s2);**

**break;**

**case 2:**

**System.out.print(s3);**

**break;**

**case 3:**

**System.out.print(s4);**

**break;**

**case 4:**

**System.out.print(s5);**

**break;**

**default:**

**break;**

**}**

**}**

**}**

ALGO-69**算法训练 字符串逆序**

时间限制：1.0s   内存限制：512.0MB

关键字：字符串

输入一个字符串，长度在100以内，按相反次序输出其中的所有字符。

样例输入

tsinghua

样例输出

auhgnist

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <string.h>**

**#include <iostream>**

**#include<functional>**

**#include <queue>**

**#include <string>**

**#include <algorithm>**

**using namespace std;**

**const int maxn = 40005;**

**int main()**

**{**

**#ifndef ONLINE\_JUDGE**

**//freopen("data.txt","r",stdin);**

**#endif**

**char str[maxn];**

**gets(str);**

**int len = strlen(str);**

**for( int i = len-1; i >= 0; i -- )**

**printf("%c",str[i]);**

**puts("");**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**char c1[100]={0},c2[100]={0},\*p1,\*p2; //初始化之后，即使后面不改，也不会再出现乱码了**

**int i=0,j=0,m;**

**gets(c1);**

**p1=c1;**

**p2=c2;**

**while (\*p1++) i++;**

**for (;;)**

**{**

**// 方法1**

**p1=c1; // 指针归位，因为现在的指针已经不指向数组首地址的位置了**

**p2=c2;**

**p2[j]=p1[i-1];**

**// 方法2**

**//c2[j] = c1[i-1]; // 比较简单的方式还是直接用数据下标**

**j++;**

**i--;**

**if (i==0) break;**

**}**

**p2[j]='\0';**

**puts(p2);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**/\*\***

**\* @param args**

**\*/**

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

**// TODO Auto-generated method stub**

**exe();**

**/\***

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str = br.readLine();**

**String [] strs = str.split("\\s+");**

**//System.out.println(strs.length);**

**if(strs[0].equals(strs[1])){**

**System.out.println(0);**

**}else{**

**int len1 = strs[0].length();**

**int len2 = strs[1].length();**

**int len = 0;**

**len = len2;**

**if(len1<len2){**

**len = len1;**

**}**

**for(int i=0;i<len;i++){**

**if(strs[0].charAt(i)==strs[1].charAt(i)){**

**continue;**

**}else{**

**System.out.println(strs[0].charAt(i)-strs[1].charAt(i));**

**break;**

**}**

**}**

**}\*/**

**}**

**public static void exe() throws IOException{**

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str = br.readLine();**

**int len = str.length();**

**for(int i=len-1;i>=0;i--){**

**System.out.print(str.charAt(i));**

**}**

**}**

**}**

ALGO-68**算法训练 判定数字**

时间限制：1.0s   内存限制：512.0MB

关键字：if-else结构，数据有效性检查

编写函数，判断某个给定字符是否为数字。

样例输入

9

样例输出

yes

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**string function(char c)**

**{**

**if(c>='0' && c<='9')**

**return "yes";**

**else return "no";**

**}**

**int main()**

**{**

**char c;**

**cin>>c;**

**cout<<function(c);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int main(int argc, char \*argv[]) {**

**char x;**

**scanf("%c",&x);**

**if('0'<=x&&x<='9')**

**{**

**printf("yes");**

**}**

**else**

**{**

**printf("no");**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main {**

**public static void main(String[] args) throws Exception {**

**Scanner scanner=new Scanner(System.in);**

**String str=scanner.nextLine();**

**boolean flag=isNumber(str);**

**if(flag){**

**System.out.println("yes");**

**} else {**

**System.out.println("no");**

**}**

**}**

**private static boolean isNumber(String str){**

**for(int i=0;i<str.length();i++){**

**if(str.charAt(i)<'0' || str.charAt(i)>'9'){**

**return false;**

**}**

**}**

**return true;**

**}**

**}**

ALGO-67**算法训练 最大值与最小值的计算**

时间限制：1.0s   内存限制：512.0MB

关键字：数组，逻辑表达式

输入11个整数，计算它们的最大值和最小值。

样例输入

0 1 2 3 4 5 6 7 8 9 10

样例输出

10 0

**参考代码：**

**C++**

**#include <iostream>**

**#include <string.h>**

**#include <string>**

**#include <algorithm>**

**#include <limits.h>**

**using namespace std;**

**int main(){**

**int min=INT\_MAX,max=INT\_MIN;**

**int t;**

**for(int i=0;i<11;i++){**

**cin>>t;**

**if(t > max)**

**max=t;**

**if(t < min)**

**min=t;**

**}**

**cout<<max<<" "<<min<<endl;**

**return 0;}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int n;**

**int a[11],i1,max=-234234,min=93759347;**

**for(i1=0;i1<11;i1++)**

**{**

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

**if(max<a[i1])**

**max=a[i1];**

**if(min>a[i1])**

**min=a[i1];**

**}**

**printf("%d %d\n",max,min);**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main{**

**public static void main(String args[]){**

**Scanner sc=new Scanner(System.in);**

**int a[]=new int[11];**

**a[0]=1000000;**

**for(int i=0;i<11;i++){**

**String s=sc.next();**

**if(s.charAt(0)>='0'&&s.charAt(0)<='9')**

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

**}**

**java.util.Arrays.sort(a);**

**if(a[10]==1000000)**

**System.out.println(1+" "+1);**

**else**

**System.out.println(a[10]+" "+a[0]);**

**}**

**}**

ALGO-66**算法训练 字符串编辑**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　从键盘输入一个字符串（长度<=40个字符），并以字符 ’.’ 结束。编辑功能有：1 D：删除一个字符，命令的方式为： D a 其中a为被删除的字符，例如：D s 表示删除字符 ’s’ ，若字符串中有多个 ‘s’，则删除第一次出现的。  
　　2 I：插入一个字符，命令的格式为：I a1 a2 其中a1表示插入到指定字符前面，a2表示将要插入的字符。例如：I s d 表示在指定字符 ’s’ 的前面插入字符 ‘d’ ，若原串中有多个 ‘s’ ，则插入在最后一个字符的前面。  
　　3 R：替换一个字符，命令格式为：R a1 a2 其中a1为被替换的字符，a2为替换的字符，若在原串中有多个a1则应全部替换。  
　　在编辑过程中，若出现被改的字符不存在时，则给出提示信息。

输入格式

　　输入文件stringedit.in共两行，第一行为原串(以’.’结束)，第二行为命令（输入方式参见“问题描述” 。

输出格式

　　输出文件stringedit.out共一行，为修改后的字符串或输出指定字符不存在的提示信息。

样例输入

This is a book.  
D s

样例输出

Thi is a book.  
D s

输入输出样例解释

　　命令为删去s，第一个在字符中出现的s在This中，即得到结果。

**参考代码：**

**C++**

**C**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**StringBuffer sb = new StringBuffer(sc.nextLine());**

**String s = sc.next();**

**if (s.equalsIgnoreCase("D")) {**

**String a = sc.next();**

**int l = sb.indexOf(a);**

**System.out.println(sb.deleteCharAt(l));**

**} else if (s.equalsIgnoreCase("I")) {**

**String a1 = sc.next();**

**String a2 = sc.next();**

**int l = sb.lastIndexOf(a1);**

**System.out.println(sb.insert(l, a2));**

**} else if (s.equalsIgnoreCase("R")) {**

**String a1 = sc.next();**

**String a2 = sc.next();**

**if (sb.indexOf(a1) == -1)**

**System.out.println("\u6307\u5b9a\u5b57\u7b26\u4e0d\u5b58\u5728");**

**else**

**System.out.println(sb.toString().replace(a1, a2));**

**}**

**}**

**}**

ALGO-65**算法训练 比赛安排**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　设有有2 n（n<=6）个球队进行单循环比赛，计划在2 n – 1天内完成，每个队每天进行一场比赛。设计一个比赛的安排，使在2n – 1天内每个队都与不同的对手比赛。

输入格式

　　输入文件matchplan.in共一行，输入n的数值。

输出格式

　　输出文件matchplan.out共（2 n – 1）行，第i行输出第i天的比赛安排。  
　　格式为：<i> A-B，C-D，……。其中i是天数，A，B分别为比赛双方的编号，每行共2 n-1个比赛场次。

样例输入

2

样例输出

<1>1-2,3-4  
<2>1-3,2-4  
<3>1-4,2-3

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**using namespace std;**

**const int N =1000;**

**bool bArrange[N][N];**

**int main()**

**{**

**int n;**

**cin>>n;**

**int row=(1<<n);**

**int volumn=1<<(n-1);**

**for(int i = 0; i< row-1 ; i++ )**

**{**

**cout<<"<"<<i+1<<">"<<"1-"<<i+2;**

**bool isArrage[N]={false};**

**isArrage[1]=isArrage[i+2]=true;**

**bArrange[1][i+2]=bArrange[i+2][1]=true;**

**for(int j =1 ; j< volumn ; j++)**

**{**

**int a;**

**for( a=2;a<=row;a++)**

**if(isArrage[a]==false)**

**break;**

**for(int b=a+1;b<=row;b++)**

**{**

**if(bArrange[a][b]==false&&isArrage[b]==false)**

**{**

**bArrange[a][b]=bArrange[b][a]=true;**

**isArrage[a]=isArrage[b]=true;**

**cout<<" "<<a<<"-"<<b;**

**break;**

**}**

**}**

**}**

**cout<<endl;**

**}**

**return 0;**

**}**

**C**

**//求最小公倍数**

**#include<stdio.h>**

**void lcm(long long a,long long b)**

**{**

**int t;**

**if(a<b)**

**{**

**t=a;**

**a=b;**

**b=t;**

**}**

**long long m,n,r;**

**m=a,n=b;**

**r=a%b;**

**while(r!=0)**

**{**

**a=b;**

**b=r;**

**r=a%b;**

**}**

**printf("%I64d\n",(m\*n)/b);**

**}**

**int main()**

**{**

**long long a,b;**

**scanf("%I64d %I64d",&a,&b);**

**if(a==0 || b==0)**

**printf("0\n");**

**else**

**lcm(a,b);**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main {**

**public static void main(String args[]){**

**Scanner sc=new Scanner(System.in);**

**int n=sc.nextInt();**

**int a[]={1,2,4,8,16,32,64,128,256,512,1024};**

**boolean b[][]=new boolean[a[n]+1][a[n]+1];**

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

**b[i][i]=true;**

**for(int i=1;i<=a[n]-1;i++){**

**boolean b1[]=new boolean[a[n]+1];**

**System.out.print("<"+i+">");**

**for(int j=1;j<=a[n];j++){**

**if(!b1[j]){**

**for(int k=j+1;k<=a[n];k++){**

**if(!b1[k]&&!b[j][k]){**

**b1[j]=true; b1[k]=true;**

**b[j][k]=b[k][j]=true;**

**System.out.print(j+"-"+k+" ");**

**break;**

**}**

**}**

**}**

**}**

**System.out.println();**

**}**

**}**

**}**

ALGO-64**算法训练 大小写判断**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个英文字母判断这个字母是大写还是小写。

输入格式

　　输入只包含一个英文字母c。

输出格式

　　如果c是大写字母，输出“upper”，否则输出“lower”。

样例输入

x

样例输出

lower

样例输入

B

样例输出

upper

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**using namespace std;**

**const int N =1000;**

**bool bArrange[N][N];**

**int main()**

**{**

**char ch;**

**cin>>ch;**

**if(islower(ch))**

**cout<<"lower";**

**else cout<<"upper";**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**char ch=0;**

**scanf("%c",&ch);**

**if(ch>='A' && ch<='Z')**

**{**

**printf("upper");**

**}**

**else**

**{**

**printf("lower");**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args) {**

**Scanner sca=new Scanner(System.in);**

**String c=sca.next();**

**char a=c.charAt(0);**

**if(a>=97&&a<=122)**

**System.out.println("lower");**

**else if(a>=65&&a<=90)**

**System.out.println("upper");**

**}}**

ALGO-63**算法训练 乘法表**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输出九九乘法表。

输出格式

　　输出格式见下面的样例。乘号用“\*”表示。

样例输出

下面给出输出的前几行：  
1\*1=1  
2\*1=2 2\*2=4  
3\*1=3 3\*2=6 3\*3=9  
4\*1=4 4\*2=8 4\*3=12 4\*4=16  
……

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**for(int i=1;i<10;i++)**

**{**

**for(int j=1;j<=i;j++)**

**{**

**cout<<i<<"\*"<<j<<"="<<i\*j<<"\t";**

**}**

**cout<<"\n";**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**int i,j;**

**for(i=1;i<10;i++)**

**{**

**for(j=1;j<=i;j++)**

**printf("%d\*%d=%d ",i,j,i\*j);**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**public class Main {**

**public static void main(String[] args) {**

**for(int i=1;i<=9;i++){**

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

**System.out.print(i+"\*"+j+"="+(i\*j)+" ");**

**}**

**System.out.println(i+"\*"+i+"="+(i\*i));**

**}**

**}**

**}**

ALGO-62**算法训练 平方计算**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入正整数a, m，输出a^2%m，其中^表示乘方，即a^2表示a的平方，%表示取余。

输入格式

　　输入包含两个整数a, m，a不超过10000。

输出格式

　　输出一个整数，即a^2%m的值。

样例输入

5 6

样例输出

1

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**// 求数组的和，其中p为指针第一个元素的指针，n为数组中的元素个数。**

**int Sum(int \*p, int n)**

**{**

**int sum=0;**

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

**sum+=p[i];**

**return sum;**

**}**

**int s[100];**

**int n;**

**int main()**

**{**

**cin >> n;**

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

**cin >> s[i];**

**cout << Sum(s, n);**

**return 0;**

**}**

**C**

**//平方计算**

**#include<stdio.h>**

**int main()**

**{**

**int a,m;**

**scanf("%d %d",&a,&m);**

**printf("%d\n",(a\*a)%m);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str [] = br.readLine().split(" ");**

**int a = Integer.parseInt(str[0]);**

**int m = Integer.parseInt(str[1]);**

**int Mi = (int)Math.pow(a, 2);**

**System.out.print(Mi%m);**

**}**

**}**

ALGO-61**算法训练 奇偶判断**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　能被2整除的数称为偶数，不能被2整除的数称为奇数。给一个整数x，判断x是奇数还是偶数。

输入格式

　　输入包括一个整数x，0<=x<=100000000。

输出格式

　　如果x是奇数，则输出“odd”，如果是偶数，则输出“even”。

样例输入

10

样例输出

even

样例输入

2009

样例输出

odd

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int x;**

**cin>>x;**

**if(x%2==0) cout<<"even";**

**else cout<<"odd";**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int main(int argc, char \*argv[]) {**

**int a[100];**

**int n;**

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

**if(n%2==0)**

**printf("even");**

**else**

**printf("odd");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**@SuppressWarnings("resource")**

**Scanner scan = new Scanner(System.in);**

**int a = scan.nextInt();**

**if (a % 2 == 0)**

**System.out.println("even");**

**else**

**System.out.println("odd");**

**}**

**}**

ALGO-60**算法训练 矩阵乘方**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个矩阵A,一个非负整数b和一个正整数m，求A的b次方除m的余数。  
　　其中一个nxn的矩阵除m的余数得到的仍是一个nxn的矩阵，这个矩阵的每一个元素是原矩阵对应位置上的数除m的余数。  
　　要计算这个问题，可以将A连乘b次，每次都对m求余，但这种方法特别慢，当b较大时无法使用。下面给出一种较快的算法(用A^b表示A的b次方)：  
　　若b=0，则A^b%m=I%m。其中I表示单位矩阵。  
　　若b为偶数，则A^b%m=(A^(b/2)%m)^2%m，即先把A乘b/2次方对m求余，然后再平方后对m求余。  
　　若b为奇数，则A^b%m=(A^(b-1)%m)\*a%m，即先求A乘b-1次方对m求余，然后再乘A后对m求余。  
　　这种方法速度较快，请使用这种方法计算A^b%m，其中A是一个2x2的矩阵，m不大于10000。

输入格式

　　输入第一行包含两个整数b, m，第二行和第三行每行两个整数，为矩阵A。

输出格式

　　输出两行，每行两个整数，表示A^b%m的值。

样例输入

2 2  
1 1  
0 1

样例输出

1 0  
0 1

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**// 坐标或向量的结构**

**struct pointer {**

**int x;**

**int y;**

**};**

**// 将A坐标按B向量平移，结果保存在C坐标中。**

**void move(pointer \*C, pointer \*A, pointer \*B)**

**{**

**C->x=A->x+B->x;**

**C->y=A->y+B->y;**

**}**

**int main()**

**{**

**pointer A, B, C;**

**cin >> A.x >> A.y;**

**cin >> B.x >> B.y;**

**move(&C, &A, &B);**

**cout << C.x << " " << C.y;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int b, m;**

**int a[2][2], ans[2][2], temp[2][2] = {1,1,1,1};**

**void play()**

**{**

**int cnt, cnt2;**

**for(cnt = 0; cnt < 2; ++cnt)**

**{**

**for(cnt2 = 0; cnt2 < 2; ++cnt2)**

**{**

**printf("%d ", ans[cnt][cnt2]);**

**}**

**printf("\n");**

**}**

**}**

**void cp(int arr1[][2], int arr2[][2])**

**{**

**int cnt, cnt2;**

**for(cnt = 0; cnt < 2 ;++cnt)**

**for(cnt2 = 0; cnt2 < 2; ++cnt2)**

**arr1[cnt][cnt2] = arr2[cnt][cnt2];**

**}**

**void mod(int arr[][2])**

**{**

**int cnt, cnt2;**

**for(cnt = 0; cnt < 2; ++cnt)**

**for(cnt2 = 0; cnt2 < 2; ++cnt2)**

**arr[cnt][cnt2] %= m;**

**}**

**void fun2(int a[][2], int b[][2])**

**{**

**int cnt, cnt2;**

**temp[0][0] = a[0][0]\*b[0][0]+a[0][1]\*b[1][0];**

**temp[0][1] = a[0][0]\*b[0][1]+a[0][1]\*b[1][1];**

**temp[1][0] = a[1][0]\*b[0][0]+a[1][1]\*b[1][0];**

**temp[1][1] = a[1][0]\*b[0][1]+a[1][1]\*b[1][1];**

**}**

**void fun(int arr[][2], int k)**

**{**

**int cnt;**

**if(k == 0)**

**{**

**mod(temp);**

**cp(ans, temp);**

**return;**

**}**

**if(k == 1)**

**{**

**mod(ans);**

**return;**

**}**

**if(k == 2)**

**{**

**fun2(a, a);**

**cp(ans, temp);**

**// printf("2\n");**

**// play();**

**mod(ans);**

**return;**

**}**

**if(k%2 == 0)**

**{**

**fun(arr, k/2);**

**fun2(ans, ans);**

**cp(ans, temp);**

**//printf("=0\n");**

**//play();**

**mod(ans);**

**return;**

**}**

**if(k%2 != 0)**

**{**

**fun(arr, k-1);**

**fun2(ans, arr);**

**cp(ans, temp);**

**//printf("!=0\n");**

**// play();**

**mod(ans);**

**return;**

**}**

**}**

**int main()**

**{**

**int cnt, cnt2;**

**scanf("%d%d", &b, &m);**

**for(cnt = 0; cnt < 2; ++cnt)**

**for(cnt2 = 0; cnt2 < 2; ++cnt2)**

**{**

**scanf("%d", &a[cnt][cnt2]);**

**ans[cnt][cnt2] = a[cnt][cnt2];**

**}**

**fun(a, b);**

**play();**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

**static int m;**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String[] str = br.readLine().split(" ");**

**int b = Integer.parseInt(str[0]);**

**m = Integer.parseInt(str[1]);**

**int[][] arr = new int[2][2];**

**for (int i = 0; i < arr.length; i++) {**

**String[] tag = br.readLine().split(" ");**

**for (int j = 0; j < arr[i].length; j++) {**

**arr[i][j] = Integer.parseInt(tag[j]);**

**}**

**}**

**int[][] tak = new int[2][2];**

**tak = mulpow(arr, b);**

**for (int i = 0; i < 2; i++) {**

**for (int j = 0; j < 2; j++) {**

**System.out.print(tak[i][j] % m + " ");**

**}**

**System.out.println();**

**}**

**}**

**public static int[][] mul(int[][] x, int[][] y) {**

**int[][] temp = new int[2][2];**

**for (int a = 0; a < 2; a++) {**

**for (int b = 0; b < 2; b++) {**

**temp[a][b] = 0;**

**for (int c = 0; c < 2; c++) {**

**temp[a][b] += x[a][c] \* y[c][b];**

**temp[a][b] %= m;**

**}**

**}**

**}**

**return temp;**

**}**

**public static int[][] mulpow(int[][] arr, int b) {**

**if (b == 0) {**

**int[][] temp = new int[2][2];**

**for (int x = 0; x < 2; x++) {**

**for (int y = 0; y < 2; y++) {**

**if (x == y)**

**temp[x][y] = 1;**

**}**

**}**

**return temp;**

**}**

**if (b == 1)**

**return arr;**

**int[][] rep = mulpow(arr, b / 2);**

**if (b % 2 == 0)**

**return mul(rep, rep);**

**else**

**return mul(mul(rep, rep), arr);**

**}**

**}**

ALGO-59**算法训练 快速排序**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　快速排序是最经常使用的一种排序方式，对于给定的n个数组成的一个数组，请使用快速排序对其进行排序。  
　　对于这道题，你只需要实现一个用于排序的函数qsort，函数的头部如下：  
　　void qsort(int array[], int size)  
　　其中array是待排序的数组，size是它的大小，注意数组中的元素是从0开始编号的，也就是说你要排序的元素是array[0..size-1]。  
　　在你提交的代码中，只需要包含这个函数。你可以加入其他的辅助函数，但不能有main函数。

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**// 交换两个整数**

**void Swap(int \*a, int \*b)**

**{**

**int temp=\*a;**

**\*a=\*b;**

**\*b=temp;**

**}**

**int main()**

**{**

**int a, b;**

**cin >> a >> b;**

**Swap(&a, &b);**

**cout << a << " " << b;**

**return 0;**

**}**

**C**

**JAVA**

ALGO-58**算法训练 字串逆序**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个字符串，将这个串的所有字母逆序后输出。

输入格式

　　输入包含一个字符串，长度不超过100，字符串中不含空格。

输出格式

　　输出包含一个字符串，为上面字符串的逆序。

样例输入

tsinsen

样例输出

nesnist

**参考代码：**

**C++**

**#include <iostream>**

**using namespace std;**

**int sum(int x, int y)**

**{**

**return x+y;**

**}**

**int main()**

**{**

**int a, b, c;**

**cin >> a >> b;**

**c = sum(a, b);**

**cout << c << endl;**

**return 0;**

**}**

**C**

**//字串逆序**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**char str[100];**

**int len;**

**scanf("%s",str);**

**len=strlen(str)-1;**

**for(;len>=0;len--)**

**printf("%c",str[len]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**while (scanner.hasNext()) {**

**String s = scanner.nextLine();**

**for (int i = s.length() - 1; i >= 0; i--) {**

**System.out.print(s.charAt(i));**

**}**

**System.out.println();**

**}**

**}**

**}**

ALGO-57**算法训练 删除多余括号**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　从键盘输入一个含有括号的四则运算表达式，要求去掉可能含有的多余的括号，结果要保持原表达式中变量和运算符的相对位置不变，且与原表达式等价,不要求化简。另外不考虑'+'　　'-'用作正负号的情况，即输入表达式不会出现(+a)或(-a)的情形。

输入格式

　　表达式字符串，长度不超过255,　　并且不含空格字符。表达式中的所有变量都是单个小写的英文字母, 运算符只有加+减-乘\*除/等运算符号。

输出格式

　　去掉多余括号后的表达式

样例输入

样例一：a+(b+c)-d样例二：a+b/(c+d)样例三：(a\*b)+c/d样例四：((a+b)\*f)-(i/j)

样例输出

样例一：a+b+c-d样例二：a+b/(c+d)样例三：a\*b+c/d样例四：(a+b)\*f-i/j

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**#include "algorithm"**

**#include "stack"**

**using namespace std;**

**int cacluExprePriority(string str,bool &hasC)**

**{**

**int left=0;**

**int right=0;**

**bool bfind=false;**

**for(int i=0;i<str.size();i++)**

**{**

**if(str[i]=='(')**

**left++;**

**if(str[i]==')')**

**right++;**

**if(str[i]=='/')**

**hasC=true;**

**if(str[i]=='\*'||str[i]=='/')**

**{**

**if(left==right)**

**return 2;**

**}**

**if(str[i]=='+'||str[i]=='-')**

**{**

**bfind=true;**

**}**

**}**

**return bfind?1:-1;**

**}**

**bool vis[1000];**

**int safe(int i,int n)**

**{**

**if(i<0)**

**return 0;**

**if(i>=n)**

**return n-1;**

**return i;**

**}**

**void findIndexOfBrackets(string str)**

**{**

**stack<int>q;**

**for(int i=0;i<str.size();i++)**

**{**

**if(str[i]=='(')**

**{**

**q.push(i);**

**}**

**if(str[i]==')')**

**{**

**int s=q.top();**

**int t=i;**

**bool hasC=false;**

**int priority=cacluExprePriority(str.substr(s+1,t-s-1),hasC);**

**q.pop();**

**bool temp=false;**

**if(s-1>=0&&cacluExprePriority(str.substr(safe(s-1,str.size()),1),temp)>=cacluExprePriority(str.substr(safe(t+1,str.size()),1),temp))**

**{**

**char op=str[s-1];**

**{**

**if(op=='+')**

**{**

**vis[s]=vis[t]=true;**

**}**

**if(op=='-')**

**{**

**if(priority==2)**

**vis[s]=vis[t]=true;**

**}**

**if(op=='\*')**

**{**

**if(priority==2&&hasC==false)**

**vis[s]=vis[t]=true;**

**}**

**}**

**}**

**else if(t+1<str.size())**

**{**

**char op=str[t+1];**

**if(op=='+'||op=='-')**

**{**

**vis[s]=vis[t]=true;**

**}**

**if(op=='\*'||op=='/')**

**{**

**if(priority==2)**

**vis[s]=vis[t]=true;**

**}**

**}**

**}**

**}**

**}**

**int main()**

**{**

**string exper;**

**cin>>exper;**

**findIndexOfBrackets(exper);**

**for(int i=0;i<exper.size();i++)**

**if(vis[i]==false)**

**cout<<exper[i];**

**cout<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int q(char \*ch)**

**{**

**int i=0,z=0;**

**ch[i] = '#';**

**while (ch[i] != ')'||z!=0)**

**{**

**if (ch[i]=='(')**

**{**

**z++;**

**}**

**if (ch[i]==')')**

**{**

**z--;**

**}**

**i++;**

**}**

**ch[i] = '#';**

**return i;**

**}**

**int f(char \*ch, char a)**

**{**

**int i = 0,jj=0;**

**if (a == '+')**

**{**

**while (ch[i] != ')')**

**{**

**if (ch[i+1] == '(')**

**if (f(&ch[i], ch[i - 1]) == 0)**

**i += q(&ch[i]);**

**else**

**{**

**while (ch[i] != ')')**

**{**

**i++;**

**}**

**}**

**i++;**

**}**

**if (ch[i + 1] == '\*' || ch[i + 1] == '/')**

**{**

**return 1;**

**}**

**else**

**{**

**return 0;**

**}**

**}**

**if (a == '-')**

**{**

**while (ch[i] != ')')**

**{**

**if (ch[i] == '(')**

**if (f(&ch[i+1], ch[i - 1]) == 0)**

**i += q(&ch[i]);**

**else**

**{**

**while (ch[i] != ')')**

**{**

**i++;**

**}**

**}**

**if (ch[i] == '+' || ch[i] == '-')**

**return 1;**

**i++;**

**}**

**if (ch[i + 1] == '\*' || ch[i + 1] == '/')**

**{**

**return 1;**

**}**

**else**

**{**

**return 0;**

**}**

**}**

**if (a == '\*')**

**{**

**while (ch[i] != ')')**

**{**

**if (ch[i] == '(')**

**if (f(&ch[i+1], ch[i - 1]) == 0)**

**q(&ch[i]);**

**else**

**{**

**while (ch[i] != ')')**

**{**

**i++;**

**}**

**}**

**if (ch[i] == '+' || ch[i] == '-')**

**return 1;**

**i++;**

**}**

**return 0;**

**}**

**if (a == '/')**

**{**

**while (ch[i] != ')')**

**{**

**if (ch[i] == '(')**

**if (f(&ch[i+1], ch[i - 1]) == 0)**

**q(&ch[i]);**

**else**

**{**

**while (ch[i] != ')')**

**{**

**i++;**

**}**

**}**

**if (ch[i] == '+' || ch[i] == '-' || ch[i] == '/' || ch[i] == '\*')**

**return 1;**

**i++;**

**}**

**}**

**while (ch[i] != ')')**

**{**

**if (ch[i] == '(')**

**if (f(&ch[i+1], ch[i - 1]) == 0)**

**i += q(&ch[i]);**

**else**

**{**

**while (ch[i] != ')')**

**{**

**i++;**

**}**

**}**

**if (ch[i] == '+' || ch[i] == '-')**

**jj=1;**

**i++;**

**}**

**if ((ch[i + 1] == '\*' || ch[i + 1] == '/')&&jj==1)**

**{**

**return 1;**

**}**

**else**

**{**

**return 0;**

**}**

**}**

**void g(char \*a)**

**{**

**int i = 0;**

**while (a[i]!='\0')**

**{**

**if (a[i] == '(')**

**if (f(&a[i + 1], a[i - 1]) == 0)**

**q(&a[i]);**

**else**

**{**

**while (a[i] != ')')**

**{**

**i++;**

**}**

**}**

**i++;**

**}**

**}**

**int main()**

**{**

**int l = 0, i = 0;**

**char ch[100];**

**scanf("%s", ch);**

**g(ch);**

**while (ch[i]!='\0')**

**{**

**if (ch[i]!='#')**

**{**

**printf("%c", ch[i]);**

**}**

**i++;**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.util.ArrayList;**

**import java.util.List;**

**public class Main{**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**char[] chs = br.readLine().toCharArray();**

**List<Character> list = new ArrayList<Character>();**

**for (int i = 0; i < chs.length; i++) {**

**list.add(chs[i]);**

**}**

**String s = "";**

**for (int i = 0; i < func(list).size(); i++) {**

**s += list.get(i);**

**}**

**System.out.println(s);**

**}**

**public static List<Character> func(List<Character> list) {**

**for (int i = 0; i < list.size(); i++) {**

**if (list.get(i) == '+' || list.get(i) == '-') {**

**if (list.get(i - 1) == ')' && list.get(i + 1) == '(') {**

**list.remove(i - 1);**

**list.remove(i);**

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

**if (list.get(j) == '(') {**

**list.remove(j);**

**break;**

**}**

**}**

**for (int k = i + 1; k < list.size(); k++) {**

**if (list.get(k) == ')') {**

**list.remove(k);**

**break;**

**}**

**}**

**}**

**if (list.get(i - 1) == ')') {**

**list.remove(i - 1);**

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

**if (list.get(j) == '(') {**

**list.remove(j);**

**break;**

**}**

**}**

**}**

**if (list.get(i + 1) == '(') {**

**if (list.get(i) == '+')**

**for (int k = i + 1; k < list.size(); k++) {**

**if (list.get(k) == ')' && !list.contains('/')**

**&& !list.contains('\*')) {**

**list.remove(k);**

**list.remove(i + 1);**

**break;**

**}**

**}**

**}**

**}**

**}**

**return list;**

**}**

**}**

ALGO-56**算法训练 邮票**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个信封，有N（1≤N≤100）个位置可以贴邮票，每个位置只能贴一张邮票。我们现在有M(M<=100)种不同邮资的邮票，面值为X1,X2….Xm分（Xi是整数，1≤Xi≤255），每种都有N张。  
  
　　显然，信封上能贴的邮资最小值是min(X1, X2, …, Xm)，最大值是 N\*max(X1, X2, …,　　Xm)。由所有贴法得到的邮资值可形成一个集合（集合中没有重复数值），要求求出这个集合中是否存在从1到某个值的连续邮资序列，输出这个序列的最大值。  
  
　　例如，N=4，M=2，面值分别为4分，1分，于是形成1，2，3，4，5，6，7，8，9，10，12，13，16的序列，而从1开始的连续邮资序列为1，2，3，4，5，6，7，8，9，10，所以连续邮资序列的最大值为10分。

输入格式

　　第一行：最多允许粘贴的邮票张数N；第二行：邮票种数M；第三行：空格隔开的M个数字，表示邮票的面值Xi。注意：Xi序列不一定是大小有序的！

输出格式

　　从1开始的连续邮资序列的最大值MAX。若不存在从1分开始的序列（即输入的邮票中没有1分面额的邮票），则输出0.

样例输入

样例一：424 1样例二：1052 4 6 8 10

样例输出

样例一：10样例二：0

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<string.h>**

**#define N 110**

**int d[N\*260],c[N];**

**int n,m,ans;**

**int main()**

**{**

**scanf("%d%d",&n,&m);**

**int i;**

**for(i=1;i<=m;i++) scanf("%d",&c[i]);**

**memset(d,0,sizeof(d));**

**ans=0;**

**while(true)**

**{**

**ans++;**

**for(i=1;i<=m;i++)**

**if(ans-c[i]>=0)**

**if(d[ans]==0 || d[ans]>d[ans-c[i]]+1)**

**d[ans]=d[ans-c[i]]+1;**

**if(d[ans]==0 || d[ans]>n)**

**{**

**printf("%d\n",ans-1);**

**break;**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<string.h>**

**int main()**

**{**

**int N,M,i,Max,zuixiaozhi,t,found;//N个位置。M种邮资**

**int youzi[260];**

**int dp[26000];**

**scanf("%d %d",&N,&M);**

**for(i=0;i<M;i++)**

**{**

**scanf("%d",&youzi[i]);**

**}**

**memset(dp,0,sizeof(dp));**

**Max=0;**

**while(1)**

**{**

**Max++;**

**found=0;**

**zuixiaozhi=10000000;**

**for(i=0;i<M;i++)**

**{**

**t=Max-youzi[i];**

**if(t>=0&&dp[t]+1<zuixiaozhi)**

**{**

**dp[Max]=dp[t]+1;**

**zuixiaozhi=dp[t]+1;**

**found=1;**

**}**

**}**

**//printf("%d\n",dp[Max]);**

**if(dp[Max]>N||found==0)**

**{**

**printf("%d\n",Max-1);**

**break;**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.LinkedList;**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt(), m = sc.nextInt();**

**int max = Integer.MIN\_VALUE;**

**LinkedList<Integer> array = new LinkedList<>();**

**for (int i = 0; i < m; ++i) {**

**int temp = sc.nextInt();**

**array.add(temp);**

**if (max < temp) {**

**max = temp;**

**}**

**}**

**int maxLen = max \* n + 1;**

**int[] x = new int[maxLen];**

**for (int i = 0; i < maxLen; ++i) {**

**x[i] = n + 1;**

**}**

**while (!array.isEmpty()) {**

**x[array.remove()] = 1;**

**}**

**for (int i = 1; i < maxLen; ++i) {**

**if (x[i] == n + 1) {**

**System.out.println(i - 1);**

**break;**

**}**

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

**int len = i + j;**

**if (len >= maxLen) {**

**break;**

**}**

**int temp = x[i] + x[j];**

**if (temp < x[len]) {**

**x[len] = temp;**

**}**

**}**

**}**

**}**

**}**

ALGO-55**算法训练 矩阵加法**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定两个N×M的矩阵，计算其和。其中：  
　　N和M大于等于1且小于等于100，矩阵元素的绝对值不超过1000。

输入格式

　　输入数据的第一行包含两个整数N、M，表示需要相加的两个矩阵的行数和列数。接下来2\*N行每行包含M个数，其中前N行表示第一个矩阵，后N行表示第二个矩阵。

输出格式

　　你的程序需要输出一个N\*M的矩阵，表示两个矩阵相加的结果。注意，输出中每行的最后不应有多余的空格，否则你的程序有可能被系统认为是Presentation　　Error

样例输入

2 21 23 45 67 8

样例输出

6 810 12

**参考代码：**

**C++**

**#include <stdio.h>**

**main()**

**{**

**int n,m;**

**int i,j;**

**int a[100][100];**

**int b[100][100];**

**int c[100][100];**

**scanf("%d %d",&n,&m);**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**scanf("%d",&a[i][j]);**

**}**

**}**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**scanf("%d",&b[i][j]);**

**}**

**}**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**c[i][j]=a[i][j]+b[i][j];**

**}**

**}**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**printf("%d\t",c[i][j]);**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main(void)**

**{**

**static int a[100][100],b[100][100];**

**int n,m;**

**int i,j;**

**scanf("%d%d",&n,&m);**

**for(i=0;i<n;i++)**

**for(j=0;j<m;j++)**

**scanf("%d",&a[i][j]);**

**for(i=0;i<n;i++)**

**for(j=0;j<m;j++)**

**scanf("%d",&b[i][j]);**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**a[i][j]+=b[i][j];**

**printf("%d ",a[i][j]);**

**}**

**printf("\n");**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**int m = sc.nextInt();**

**int[][] a = new int[n][m];**

**int i, j;**

**for (i = 0; i < n; i++)**

**for (j = 0; j < m; j++)**

**a[i][j] = sc.nextInt();**

**for (i = 0; i < n; i++)**

**for (j = 0; j < m; j++)**

**a[i][j] += sc.nextInt();**

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

**for (j = 0; j < m - 1; j++)**

**System.out.print(a[i][j] + " ");**

**System.out.println(a[i][j]);**

**}**

**}**

**}**

ALGO-54**算法训练 简单加法(基本型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　首先给出简单加法算式的定义：  
　　如果有一个算式(i)+(i+1)+(i+2),(i>=0)，在计算的过程中，没有任何一个数位出现了进位，则称其为简单的加法算式。  
　　例如：i=3时，3+4+5=12，有一个进位，因此3+4+5不是一个简单的加法算式；又如i=112时，112+113+114=339，没有在任意数位上产生进位，故112+113+114是一个简单的加法算式。  
  
　　问题：给定一个正整数n，问当i大于等于0且小于n时,有多少个算式(i)+(i+1)+(i+2)是简单加法算式。其中n<10000。

输入格式

　　一个整数，表示n

输出格式

　　一个整数,表示简单加法算式的个数

样例输入

4

样例输出

3

**参考代码：**

**C++**

**#include<iostream>**

**using namespace std;**

**int f(int n)**

**{**

**if((3\*(n%10)+3)>=10)**

**{**

**return 0;**

**}**

**n/=10;**

**while(n)**

**{**

**if(3\*(n%10)>=10)**

**{**

**return 0;**

**}**

**n/=10;**

**}**

**return 1;**

**}**

**int main()**

**{**

**int n,cnt=0;**

**int i;**

**cin>>n;**

**for(i=0;i<n;i++)**

**{**

**if(f(i))**

**cnt++;**

**}**

**cout<<cnt;**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main(void)**

**{**

**int n,i;**

**int count=0;**

**int a,b,c,d;**

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

**for(i=0;i<n;i++)**

**{**

**d=i%10;**

**c=i/10%10;**

**b=i/100%10;**

**a=i/1000;**

**if(a<=2&&b<=2&&c<=2&&d<=2)**

**count++;**

**}**

**printf("%d",count);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args) {**

**Scanner sca=new Scanner(System.in);**

**int n=sca.nextInt();**

**int i=0;int p=0;**

**for(;i<n;i++){**

**int a=i;**

**int b=i+1;**

**int c=i+2;**

**String d=Integer.toString(a);**

**String e=Integer.toString(b);**

**String f=Integer.toString(c);**

**int g=d.charAt(d.length()-1)-48;**

**int h=e.charAt(e.length()-1)-48;**

**int j=f.charAt(f.length()-1)-48;**

**int sum=g+h+j;**

**if(sum<10){**

**p++;}**

**}**

**System.out.println(p);**

**}**

**}**

ALGO-52**算法训练 排列问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　求一个0～N-1的排列（即每个数只能出现一次），给出限制条件（一张N\*N的表，第i行第j列的1或0，表示为j-1这个数不能出现在i-1这个数后面，并保证第i行第i列为0），将这个排列看成一个自然数，求从小到大排序第K个排列。

数据规模和约定

　　N<=10，K<=500000

输入格式

　　第一行为N和K,接下来的N行，每行N个数，0表示不能，1表示能

输出格式

　　所求的排列

样例输入

3 20 1 11 0 00 1 0

样例输出

1 0 2解释：对于N=3的没有任何限制的情况第一：0 1 2第二：0 2 1第三：1 0 2第四：1 2 0第五：2 0 1第六：2 1 0根据题目所给的限制条件由于2不能出现在1后面，0不能出现在2后面第一：0 2 1第二：1 0 2第三：2 1 0

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**#include "algorithm"**

**#include "stack"**

**using namespace std;**

**const int N =101;**

**int a[11];**

**bool map[11][11];**

**bool isOk(int n)**

**{**

**for(int i=0;i<n-1;i++)**

**if(map[ a[i] ][ a[i+1] ] == 0)**

**{**

**return false;**

**}**

**return true;**

**}**

**void init()**

**{**

**for(int i=0;i<10;i++)**

**a[i]=i;**

**}**

**int main()**

**{**

**int n;**

**int k;**

**cin>>n>>k;**

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

**for(int j = 0; j < n; j++)**

**cin>>map[i][j];**

**init();**

**int num=0;**

**do**

**{**

**if(isOk(n))**

**num++;**

**if(num==k)**

**{**

**cout<<a[0];**

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

**cout<<" "<<a[i];**

**cout<<endl;**

**break;**

**}**

**//cout<<endl;**

**} while (std::next\_permutation(a,a+n));**

**}**

**C**

**JAVA**

**import java.util.\*;**

**public class Main {**

**public static void main(String args[]) {**

**Scanner sc=new Scanner(System.in);**

**String[] str=new String[100];**

**int count=0;**

**int sum=0;**

**int n=sc.nextInt();**

**int m=sc.nextInt();**

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

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

**int w=sc.nextInt();**

**if(i!=j&&w==0)**

**str[++count]=(i-1)+""+(j-1);**

**}**

**}**

**int a[]={0,1,2,3,4,5,6,7,8,9};**

**int date[]={1,1,2,6,24,120,720,5040,40320,362880};**

**boolean b[] = new boolean[10];**

**oter:for(int i=0;i<date[a[n]];i++){**

**String s="";**

**int r = i;**

**int num = 0;**

**for(int j=0;j<n;j++)**

**b[j]=false;**

**for (int j = n; j >=1; j--) {**

**num=0;**

**int x = r / date[j-1];**

**r = r % date[j-1];**

**for (int k = 0; k < n; k++) {**

**if (!b[k])**

**num++;**

**if (num == x + 1) {**

**//System.out.print(a[k]);**

**s+=a[k];**

**b[k] = true;**

**break;**

**}**

**}**

**}**

**for(int j=1;j<=count;j++){**

**if(s.contains(str[j]))**

**break;**

**if(j==count)**

**sum++;**

**}**

**if(sum==m){**

**char[] c1=s.toCharArray();**

**for(int j=0;j<n;j++)**

**System.out.print(c1[j]+" ");**

**break oter;**

**}**

**}**

**}**

**}**

ALGO-50**算法训练 数组查找及替换**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定某整数数组和某一整数b。要求删除数组中可以被b整除的所有元素，同时将该数组各元素按从小到大排序。如果数组元素数值在A到Z的ASCII之间，替换为对应字母。元素个数不超过100，b在1至100之间。

输入格式

　　第一行为数组元素个数和整数b  
　　第二行为数组各个元素

输出格式

　　按照要求输出

样例输入

7 277 11 66 22 44 33 55

样例输出

11 33 55 M

**参考代码：**

**C++**

**#include<iostream>**

**#include<algorithm>**

**using namespace std;**

**int arr[100];**

**int main()**

**{**

**int n,b;**

**int i,j;**

**cin>>n>>b;**

**for(i=0;i<n;i++)**

**{**

**cin>>arr[i];**

**if(arr[i]%b==0)**

**arr[i]=0;**

**}**

**sort(arr,arr+n);**

**for(i=0;i<n;i++)**

**{**

**if(arr[i])**

**{**

**if(arr[i]>=65 &&arr[i]<=90)**

**{**

**cout<<char(arr[i])<<" ";**

**}**

**else**

**cout<<arr[i]<<" ";**

**}**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**void del(int a[],int \*len,int m)**

**{ int i,j;**

**for(i=0;i<\*len;i++)**

**if(a[i]%m==0)**

**{**

**for(j=i;j<\*len;j++)**

**a[j]=a[j+1];**

**(\*len)--;**

**i--;**

**}**

**}**

**int cmp(const void \*a,const void \*b)**

**{**

**return \*(int \*)a-\*(int \*)b;**

**}**

**int main()**

**{**

**int a[101]={},i,m,n;**

**scanf("%d%d",&n,&m);**

**for(i=0;i<n;i++)**

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

**del(a,&n,m);**

**qsort(a,n,sizeof(a[0]),cmp);**

**for(i=0;i<n;i++)**

**{**

**if(a[i]>=65&&a[i]<=90)**

**printf("%c ",a[i]);**

**else**

**printf("%d ",a[i]);**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.List;**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**while (scanner.hasNext()) {**

**int n = scanner.nextInt();**

**int b = scanner.nextInt();**

**List<Integer> nums = new ArrayList<>();**

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

**int temp = scanner.nextInt();**

**if (temp % b != 0) {**

**nums.add(temp);**

**}**

**}**

**Collections.sort(nums);**

**for (int i = 0; i < nums.size(); i++) {**

**if (nums.get(i) >= 'A' && nums.get(i) <= 'Z') {**

**int temp = nums.get(i);**

**char ch = (char) temp;**

**System.out.print(ch);**

**} else {**

**System.out.print(nums.get(i));**

**}**

**System.out.print(i == nums.size() - 1 ? "\r\n" : " ");**

**}**

**}**

**}**

**}**

ALGO-47**算法训练 蜜蜂飞舞**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　“两只小蜜蜂呀，飞在花丛中呀……”  
  
　　话说这天天上飞舞着两只蜜蜂，它们在跳一种奇怪的舞蹈。用一个空间直角坐标系来描述这个世界，那么这两只蜜蜂初始坐标分别为(x1,y1,z1)，(x2,y2,z2)　　。在接下来它们将进行n次飞行，第i次飞行两只蜜蜂分别按照各自的速度向量飞行ti个单位时间。对于这一现象，玮玮已经观察了很久。他很想知道在蜜蜂飞舞结束时，两只蜜蜂的距离是多少。现在他就求教于你，请你写一个程序来帮他计算这个结果。

输入格式

　　第一行有且仅有一个整数n，表示两只蜜蜂将进行n次飞行。  
  
　　接下来有n行。  
  
　　第i行有7个用空格分隔开的整数ai,bi,ci,di,ei,fi,ti　　，表示第一只蜜蜂单位时间的速度向量为(ai,bi,ci) ，第二只蜜蜂单位时间的速度向量为(di,ei,fi) ，它们飞行的时间为ti 。  
  
　　最后一行有6个用空格分隔开的整数x1,y1,z1,x2,y2,z2，如题所示表示两只蜜蜂的初始坐标。

输出格式

　　输出仅包含一行，表示最后两只蜜蜂之间的距离。保留4位小数位。

样例输入

Sample 111 1 1 1 -1 1 23 0 1 2 0 0Sample 231 1 1 1 -1 1 22 1 2 0 -1 -1 22 0 0 -1 1 1 33 0 1 2 0 0

样例输出

Sample 14.2426Sample 215.3948

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cmath>**

**using namespace std;**

**int main(){**

**int x1=0,y1=0,z1=0,x2=0,y2=0,z2=0;**

**int a1,b1,c1,a2,b2,c2;**

**int t;**

**int n;**

**double s;**

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

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

**scanf("%d%d%d%d%d%d%d",&a1,&b1,&c1,&a2,&b2,&c2,&t);**

**x1+=a1\*t;**

**x2+=a2\*t;**

**y1+=b1\*t;**

**y2+=b2\*t;**

**z1+=c1\*t;**

**z2+=c2\*t;**

**}**

**scanf("%d%d%d%d%d%d",&a1,&b1,&c1,&a2,&b2,&c2);**

**x1+=a1;**

**x2+=a2;**

**y1+=b1;**

**y2+=b2;**

**z1+=c1;**

**z2+=c2;**

**s=sqrt((double)(abs(x2-x1)\*abs(x2-x1)+abs(y2-y1)\*abs(y2-y1)+abs(z2-z1)\*abs(z2-z1)));**

**printf("%.4lf\n",s);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <math.h>**

**int main()**

**{**

**int n;**

**double x1=0,y1=0,z1=0,x2=0,y2=0,z2=0;**

**int ai,bi,ci,di,ei,fi,ti;**

**int i;**

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

**for(i=0;i<n;i++)**

**{**

**scanf("%d%d%d%d%d%d%d",&ai,&bi,&ci,&di,&ei,&fi,&ti);**

**x1+=ai\*ti;**

**y1+=bi\*ti;**

**z1+=ci\*ti;**

**x2+=di\*ti;**

**y2+=ei\*ti;**

**z2+=fi\*ti;**

**}**

**scanf("%d%d%d%d%d%d",&ai,&bi,&ci,&di,&ei,&fi);**

**x1+=ai;**

**y1+=bi;**

**z1+=ci;**

**x2+=di;**

**y2+=ei;**

**z2+=fi;**

**printf("%.4lf",sqrt((x1-x2)\*(x1-x2)+(y1-y2)\*(y1-y2)+(z1-z2)\*(z1-z2)));**

**return 0;**

**}**

**JAVA**

**import java.util.ArrayList;**

**import java.util.Collections;**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**int n = sc.nextInt();**

**int x1 = 0, y1 = 0, z1 = 0;**

**int x2 = 0, y2 = 0, z2 = 0;**

**for (; n > 0; --n) {**

**int a1 = sc.nextInt(), b1 = sc.nextInt(), c1 = sc.nextInt();**

**int a2 = sc.nextInt(), b2 = sc.nextInt(), c2 = sc.nextInt();**

**int t = sc.nextInt();**

**x1 += a1 \* t;**

**y1 += b1 \* t;**

**z1 += c1 \* t;**

**x2 += a2 \* t;**

**y2 += b2 \* t;**

**z2 += c2 \* t;**

**}**

**x1 += sc.nextInt();**

**y1 += sc.nextInt();**

**z1 += sc.nextInt();**

**x2 += sc.nextInt();**

**y2 += sc.nextInt();**

**z2 += sc.nextInt();**

**x1 -= x2;**

**y1 -= y2;**

**z1 -= z2;**

**System.out.printf("%.4f", Math.sqrt(x1 \* x1 + y1 \* y1 + z1 \* z1));**

**}**

**}**

ALGO-46**算法训练 Hanoi问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　如果将课本上的Hanoi塔问题稍做修改：仍然是给定N只盘子，3根柱子，但是允许每次最多移动相邻的M只盘子（当然移动盘子的数目也可以小于M）,最少需要多少次？  
　　例如N=5，M=2时，可以分别将最小的2个盘子、中间的2个盘子以及最大的一个盘子分别看作一个整体，这样可以转变为N=3，M=1的情况，共需要移动7次。

输入格式

　　输入数据仅有一行，包括两个数N和M（0<=M<=N<=8）

输出格式

　　仅输出一个数，表示需要移动的最少次数

样例输入

5 2

样例输出

7

**参考代码：**

**C++**

**#include <iostream>**

**#include <cmath>**

**using namespace std;**

**int main()**

**{**

**int n,m;**

**cin>>n>>m;**

**if(n%m==0)**

**n/=m;**

**else n=n/m+1;**

**cout<<pow(2,n)-1;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int N,M;**

**scanf("%d%d",&N,&M);**

**N=(N+1)/M;**

**printf("%d",(1<<N)-1);**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main{**

**public static void main(String args[]){**

**Scanner sc=new Scanner(System.in);**

**int n=sc.nextInt();**

**int m=sc.nextInt();**

**int k=0;**

**if(n%m==0)**

**k=n/m;**

**else**

**k=(n+1)/m;**

**int count=1;**

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

**count\*=2;**

**System.out.println(count-1);**

**}**

**}**

ALGO-45**算法训练 调和数列问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入一个实数x，求最小的n使得，1/2+1/3+1/4+...+1/(n+1)>=x。  
  
　　输入的实数x保证大于等于0.01，小于等于5.20，并且恰好有两位小数。你的程序要能够处理多组数据，即不停地读入x，如果x不等于0.00，则计算答案，否则退出程序。  
  
　　输出格式为对于一个x，输出一行n card(s)。其中n表示要计算的答案。

输入格式

　　分行输入x的具体数值

输出格式

　　分行输出n的数值，格式为n card(s)

样例输入

1.003.710.045.190.00

样例输出

3 card(s)61 card(s)1 card(s)273 card(s)

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**#include "algorithm"**

**#include "stack"**

**#include "math.h"**

**using namespace std;**

**const int N =101;**

**double fen(int i)**

**{**

**return 1.0/i;**

**}**

**int main()**

**{**

**double x;**

**while(cin>>x, x!=0.00)**

**{**

**double ans=0;**

**for(int i=2;;i++)**

**{**

**ans+=fen(i);**

**if(ans>=x)**

**{**

**cout<<i-1<<" "<<"card(s)"<<endl;**

**break;**

**}**

**}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main(void)**

**{**

**int i;**

**double n,k,s;**

**scanf("%lf",&n);**

**while(n!=0)**

**{**

**s=0;**

**for(i=2;1;i++)**

**{**

**k=1.0/i;**

**s+=k;**

**if(s>=n)**

**{**

**printf("%d card(s)\n",i-1);**

**break;**

**}**

**}**

**scanf("%lf",&n);**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main{**

**public static void main(String args[]){**

**Scanner sc=new Scanner (System.in);**

**while(true){**

**double d=sc.nextDouble();**

**double sum=0;**

**int count=0;**

**if(d==0.00)**

**break;**

**else{**

**for(int i=2;;i++){**

**sum+=1.0/i;**

**if(sum>=d){**

**count=i-1;**

**break;**

**}**

**}**

**System.out.println(count+" card(s)");**

**}**

**}**

**}**

**}**

ALGO-40**算法训练 会议中心**

时间限制：2.0s   内存限制：512.0MB

关键字：APIO 2009

会议中心　　Siruseri政府建造了一座新的会议中心。许多公司对租借会议中心的会堂很感兴趣，他们希望能够在里面举行会议。  
　　对于一个客户而言，仅当在开会时能够独自占用整个会堂，他才会租借会堂。会议中心的销售主管认为：最好的策略应该是将会堂租借给尽可能多的客户。显然，有可能存在不止一种满足要求的策略。  
　　例如下面的例子。总共有4个公司。他们对租借会堂发出了请求，并提出了他们所需占用会堂的起止日期（如下表所示）。

|  |  |  |
| --- | --- | --- |
|  | 开始日期 | 结束日期 |
| 公司1 | 4 | 9 |
| 公司2 | 9 | 11 |
| 公司3 | 13 | 19 |
| 公司4 | 10 | 17 |

　　上例中，最多将会堂租借给两家公司。租借策略分别是租给公司1和公司3，或是公司2和公司3，也可以是公司1和公司4。注意会议中心一天最多租借给一个公司，所以公司1和公司2不能同时租借会议中心，因为他们在第九天重合了。  
　　销售主管为了公平起见，决定按照如下的程序来确定选择何种租借策略：首先，将租借给客户数量最多的策略作为候选，将所有的公司按照他们发出请求的顺序编号。对于候选策略，将策略中的每家公司的编号按升序排列。最后，选出其中字典序最小[1]的候选策略作为最终的策略。  
　　例中，会堂最终将被租借给公司1和公司3：3个候选策略是{(1,3),(2,3),(1,4)}。而在字典序中(1,3)<(1,4)<(2,3)。  
　　你的任务是帮助销售主管确定应该将会堂租借给哪些公司。

输入格式

　　输入的第一行有一个整数N，表示发出租借会堂申请的公司的个数。第2到第N+1行每行有2个整数。第i+1行的整数表示第i家公司申请租借的起始和终止日期。对于每个公司的申请，起始日期为不小于1的整数，终止日期为不大于109的整数。

输出格式

　　输出的第一行应有一个整数M，表示最多可以租借给多少家公司。第二行应列出M个数，表示最终将会堂租借给哪些公司。

数据规模和约定

　　对于50%的输入，N≤3000。在所有输入中，N≤200000。

样例输入

4  
4 9  
9 11  
13 19  
10 17

样例输出

2  
1 3  
  
[1] 字典序指在字典中排列的顺序，如果序列l1是序列l2的前缀，或者对于l1和l2的第一个不同位置j，l1[j]<l2[j]，则l1比l2小。

**参考代码：**

**C++**

**#include <cstdio>**

**#include <cstdlib>**

**#include <algorithm>**

**#include <string>**

**#include <set>**

**const char fi[] = "convention.in";**

**const char fo[] = "convention.out";**

**const int maxN = 200010;**

**const int MAX = 0x3f3f3f3f,MIN = ~MAX;**

**struct Seg**

**{**

**int L,R;**

**Seg()**

**{**

**}**

**Seg(int L,int R): L(L),R(R)**

**{**

**}**

**bool operator<(const Seg &b) const**

**{**

**return L < b.L || L == b.L && R < b.R;**

**}**

**};**

**std::set <Seg> S;**

**std::set <Seg>::iterator iter;**

**Seg req[maxN],seg[maxN],tmp[maxN];**

**int tab[maxN << 1],next[20][maxN << 1];**

**int n,cnt,Lim = 1,logLim;**

**void init\_file()**

**{**

**return;**

**}**

**inline int getint()**

**{**

**int res = 0; char tmp;**

**while(!isdigit(tmp = getchar()));**

**do res = (res << 3) + (res << 1) + tmp - '0';**

**while(isdigit(tmp = getchar()));**

**return res;**

**}**

**void readdata()**

**{**

**n = getint();**

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

**{**

**int L = getint(),R = getint();**

**req[i] = Seg(L,R);**

**tab[i << 1] = L;**

**tab[(i << 1) + 1] = R;**

**}**

**return;**

**}**

**int plc(int x)**

**{**

**for(int L = 0,R = Lim - 1; L < R + 1;)**

**{**

**int Mid = (L + R) >> 1;**

**if(x == tab[Mid]) return Mid + 1;**

**if(x < tab[Mid]) R = Mid - 1;**

**else L = Mid + 1;**

**}**

**}**

**bool cmp(const Seg &a,const Seg &b)**

**{**

**return a.R < b.R || a.R == b.R && a.L > b.L;**

**}**

**void discrete()**

**{**

**std::sort(tab,tab + (n << 1));**

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

**if(tab[i] != tab[i - 1])**

**tab[Lim++] = tab[i];**

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

**tmp[i] = req[i] = Seg(plc(req[i].L),**

**plc(req[i].R));**

**std::sort(tmp,tmp + n,cmp);**

**//这里必须要用一个临时数组，**

**//保证左界右界同时单调递增。**

**int p = 0; seg[cnt++] = tmp[0];**

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

**if(tmp[i].L > tmp[p].L)**

**seg[cnt++] = tmp[p = i];**

**return;**

**}**

**void next\_set()**

**{**

**int p = cnt; next[0][Lim + 1] = MAX;**

**for(int j = Lim; j; --j)**

**if(p > -1 && j == seg[p - 1].L)**

**next[0][j] = seg[--p].R + 1;**

**else next[0][j] = next[0][j + 1];**

**for(int i = 0;; ++i)**

**{**

**bool flag = 0;**

**next[i + 1][Lim + 1] = MAX;**

**for(int k = 1; k < Lim + 1; ++k)**

**{**

**if(next[i][k] == MAX)**

**next[i + 1][k] = MAX;**

**else next[i + 1][k] = next[i][next[i][k]];**

**if(next[i + 1][k] < MAX) flag = 1;**

**}**

**if(!flag)**

**{**

**logLim = i; break;**

**}**

**}**

**return;**

**}**

**int max\_time(int L,int R)**

**{**

**if(L > R++) return 0;**

**int ans = 0,p = L;**

**for(int i = logLim; i > -1 && p < R; --i)**

**if(next[i][p] <= R)**

**{**

**p = next[i][p]; ans += 1 << i;**

**}**

**return ans;**

**}**

**bool query(int i)**

**{**

**int L = req[i].L,R = req[i].R;**

**iter = S.lower\_bound(Seg(L,MAX));**

**if(iter-- == S.begin()) return 0;**

**if(iter->L > L || iter->R < R)**

**return 0;**

**int L1 = iter->L,R1 = iter->R;**

**if(max\_time(L1,L - 1)**

**+ max\_time(R + 1,R1)**

**+ 1 < max\_time(L1,R1))**

**//这里要满足放进去过后不影响总的答案。**

**return 0;**

**S.erase(iter);**

**if(L1 < L) S.insert(Seg(L1,L - 1));**

**if(R < R1) S.insert(Seg(R + 1,R1));**

**return 1;**

**}**

**void work()**

**{**

**printf("%d\n",max\_time(1,Lim));**

**S.insert(Seg(1,Lim));**

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

**if(query(i))**

**printf("%d ",i + 1);**

**printf("\n");**

**return;**

**}**

**int main()**

**{**

**init\_file();**

**readdata();**

**discrete();**

**next\_set();**

**work();**

**return 0;**

**}**

**C**

**JAVA**

ALGO-44**算法训练 采油区域**

时间限制：2.0s   内存限制：512.0MB

关键字：APIO 2009

采油区域　　Siruseri政府决定将石油资源丰富的Navalur省的土地拍卖给私人承包商以建立油井。被拍卖的整块土地为一个矩形区域，被划分为M×N个小块。  
　　Siruseri地质调查局有关于Navalur土地石油储量的估测数据。这些数据表示为M×N个非负整数，即对每一小块土地石油储量的估计值。  
　　为了避免出现垄断，政府规定每一个承包商只能承包一个由K×K块相连的土地构成的正方形区域。  
　　AoE石油联合公司由三个承包商组成，他们想选择三块互不相交的K×K的区域使得总的收益最大。  
　　例如，假设石油储量的估计值如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 8 | 8 | 8 | 8 | 8 | 1 | 1 | 1 |
| 1 | 8 | 8 | 8 | 8 | 8 | 1 | 1 | 1 |
| 1 | 8 | 8 | 8 | 8 | 8 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 8 | 8 | 8 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 8 | 8 | 8 |
| 1 | 1 | 1 | 1 | 1 | 1 | 9 | 9 | 9 |
| 1 | 1 | 1 | 1 | 1 | 1 | 9 | 9 | 9 |

　　如果K = 2, AoE公司可以承包的区域的石油储量总和为100, 如果K = 3, AoE公司可以承包的区域的石油储量总和为208。  
　　AoE公司雇佣你来写一个程序，帮助计算出他们可以承包的区域的石油储量之和的最大值。

输入格式

　　输入第一行包含三个整数M, N, K，其中M和N是矩形区域的行数和列数，K是每一个承包商承包的正方形的大小（边长的块数）。接下来M行，每行有N个非负整数表示这一行每一小块土地的石油储量的估计值。

输出格式

　　输出只包含一个整数，表示AoE公司可以承包的区域的石油储量之和的最大值。

数据规模和约定

　　数据保证K≤M且K≤N并且至少有三个K×K的互不相交的正方形区域。其中30%的输入数据，M, N≤ 12。所有的输入数据, M, N≤ 1500。每一小块土地的石油储量的估计值是非负整数且≤ 500。

样例输入

9 9 3  
1 1 1 1 1 1 1 1 1  
1 1 1 1 1 1 1 1 1  
1 8 8 8 8 8 1 1 1  
1 8 8 8 8 8 1 1 1  
1 8 8 8 8 8 1 1 1  
1 1 1 1 8 8 8 1 1  
1 1 1 1 1 1 8 8 8  
1 1 1 1 1 1 9 9 9  
1 1 1 1 1 1 9 9 9

样例输出

208

**参考代码：**

**C++**

**#include <cstdio>**

**#include <cstdlib>**

**#include <algorithm>**

**#include <string>**

**#define max(a, b) ((a) > (b) ? (a) : (b))**

**#define max3(a, b, c) (max((a), max(b, c)))**

**const char fi[] = "oil.in";**

**const char fo[] = "oil.out";**

**const int maxN = 1510;**

**const int MAX = 0x3f3f3f3f;**

**const int MIN = ~MAX;**

**int sum[maxN][maxN];**

**int zoxx[maxN][maxN]; // Left and below.**

**int zouh[maxN][maxN]; // Left and above.**

**int yzxx[maxN][maxN]; // Right and below.**

**int yzuh[maxN][maxN]; // Right and above.**

**int hh[maxN]; // Horizon.**

**int lp[maxN]; // Vertical.**

**int n,m,K;**

**void init\_file()**

**{**

**return;**

**}**

**inline int getint()**

**{**

**int res = 0; char tmp;**

**while(!isdigit(tmp = getchar()));**

**do res = (res << 3) + (res << 1) + tmp - '0';**

**while(isdigit(tmp = getchar()));**

**return res;**

**}**

**void readdata()**

**{**

**n = getint(); m = getint(); K = getint();**

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

**for(int j = 1; j < m + 1; ++j)**

**(sum[i][j] = getint()) += sum[i][j - 1];**

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

**for(int j = 1; j < m + 1; ++j)**

**sum[i][j] += sum[i - 1][j];**

**//预处理出二维的前缀和。**

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

**for(int j = K; j < m + 1; ++j)**

**{**

**zouh[i][j] = sum[i][j]**

**+ sum[i - K][j - K]**

**- sum[i][j - K]**

**- sum[i - K][j];**

**hh[i] = max(hh[i],zouh[i][j]); //Horizon.**

**lp[j] = max(lp[j],zouh[i][j]); //Vertical.**

**zouh[i][j] = max3(zouh[i][j],**

**zouh[i - 1][j],**

**zouh[i][j - 1]);**

**} // Left and above.**

**for(int i = n - K + 1; i; --i)**

**for(int j = m - K + 1; j; --j)**

**yzxx[i][j] = max3(sum[i - 1][j - 1]**

**+ sum[i + K - 1][j + K - 1]**

**- sum[i - 1][j + K - 1]**

**- sum[i + K - 1][j - 1],**

**yzxx[i + 1][j],**

**yzxx[i][j + 1]);**

**//Right and below.**

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

**for(int j = m - K + 1; j; --j)**

**yzuh[i][j] = max3(sum[i][j + K - 1]**

**+ sum[i - K][j - 1]**

**- sum[i][j - 1]**

**- sum[i - K][j + K - 1],**

**yzuh[i - 1][j],**

**yzuh[i][j + 1]);**

**//Right and above.**

**for(int i = n - K + 1; i; --i)**

**for(int j = K; j < m + 1; ++j)**

**zoxx[i][j] = max3(sum[i - 1][j - K]**

**+ sum[i + K - 1][j]**

**- sum[i - 1][j]**

**- sum[i + K - 1][j - K],**

**zoxx[i + 1][j],**

**zoxx[i][j - 1]);**

**//Left and below**

**return;**

**}**

**void work()**

**{**

**int ans = 0;**

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

**ans = max(ans,yzuh[i][1]**

**+ hh[i + K]**

**+ yzxx[i + K + 1][1]);**

**// Case 1.**

**for(int j = K; j < m - (K << 1); ++j)**

**ans = max(ans,zoxx[1][j]**

**+ lp[j + K]**

**+ yzxx[1][j + K + 1]);**

**// Case 2.**

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

**for(int j = K; j < m - K + 1; ++j)**

**{**

**ans = max3(ans,zouh[i][j]**

**+ yzuh[i][j + 1]**

**+ yzxx[i + 1][1], //Case 3.**

**yzuh[i][1]**

**+ zoxx[i + 1][j]**

**+ yzxx[i + 1][j + 1]); //Case 4.**

**ans = max3(ans,zouh[i][j]**

**+ zoxx[i + 1][j]**

**+ yzxx[1][j + 1], //Case 5.**

**zoxx[1][j]**

**+ yzuh[i][j + 1]**

**+ yzxx[i + 1][j + 1]); //Case 6.**

**}**

**printf("%d\n",ans);**

**return;**

**}**

**int main()**

**{**

**init\_file();**

**readdata();**

**work();**

**return 0;**

**}**

**#undef max**

**C**

**JAVA**

ALGO-43**算法训练 A+B Problem**

时间限制：1.0s   内存限制：512.0MB

关键字：试用 C++ 入门

问题描述

　　输入A,B。  
　　输出A+B。

输入格式

　　输入包含两个整数A,B，用一个空格分隔。

输出格式

　　输出一个整数，表示A+B的值。

样例输入

5 8

样例输出

13

数据规模和约定

　　-1,000,000,000<=A,B<=1,000,000,000。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int a,b;**

**cin>>a;**

**cin>>b;**

**cout<<a+b;**

**cout<<'\n';**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int sum(int a,int b)**

**{**

**return a+b;**

**}**

**int main()**

**{**

**int a,b;**

**scanf("%d%d",&a,&b);**

**//sum(a,b);**

**printf("%d",sum(a,b));**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str[] = br.readLine().split(" ");**

**int n = Integer.parseInt(str[0]);**

**int m = Integer.parseInt(str[1]);**

**System.out.print(n+m);**

**}**

**}**

ALGO-41**算法训练 弹弹堂**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　XX无聊玩弹弹堂，战斗力太低啦！

输入格式

　　测试数据的输入一定会满足的格式。  
　　例：输入的第一行包含两个整数n, m，分别表示矩阵的行数和列数。接下来n行，每行m个正整数，表示输入的矩阵。

输出格式

　　要求用户的输出满足的格式。  
　　例：输出1行，包含一个整数，表示矩阵中所有元素的和。

样例输入

一个满足题目要求的输入范例。  
例：  
2 2  
1 2  
3 4

样例输出

与上面的样例输入对应的输出。  
例：  
10

数据规模和约定

　　输入数据中每一个数的范围。  
　　例：0<n,m<100, 0<=矩阵中的每个数<=1000。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**double sum=1.54545;**

**int m=0,n=0,t;**

**int i=0,j=0;**

**scanf("%d%d",&m,&n);**

**for(i=0;i<m;i++)**

**{**

**for(j=0;j<n;j++)**

**{**

**scanf("%d",&t);**

**sum=sum+t;**

**}**

**}**

**cout<<sum;**

**cout<<'\n';**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int main()**

**{**

**printf("1.54545");**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main**

**{**

**public static void main(String[] args)**

**{**

**System.out.println("1.54545");**

**return;**

**}**

**}**

ALGO-39**算法训练 数组排序去重**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入10个整数组成的序列，要求对其进行升序排序，并去掉重复元素。

输入格式

　　10个整数。

输出格式

　　多行输出，每行一个元素。

样例输入

2 2 3 3 1 1 5 5 5 5

样例输出

1  
2  
3  
5

**参考代码：**

**C++**

**#include <vector>**

**#include <iostream>**

**#include <algorithm>**

**using namespace std;**

**int main()**

**{**

**int i,t;**

**vector<int> v;**

**for (i = 0; i < 10; i++)**

**{**

**cin>>t;**

**v.push\_back(t);**

**}**

**sort(v.begin(),v.end());**

**v.erase(unique(v.begin(),v.end()),v.end());//unique(n.begin(),n.end())；unique()把相邻元素重复的甩到后面//erase()用来删掉后面的重复元素.//一定要先排序，再使用unique()，**

**for (i = 0; i < v.size(); i++)**

**cout<<v[i]<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**int A[10];**

**int com(const void \*a, const void \*b)**

**{**

**return \*(int\*)a-\*(int\*)b;**

**}**

**int main()**

**{**

**int i;**

**int last;**

**for(i=0;i<10;i++)**

**scanf("%d",&A[i]);**

**qsort(A,10,4,com);**

**for(i=0;i<10;i++)**

**if(i)**

**{**

**if(last!=A[i])**

**{**

**printf("%d\n",A[i]);**

**last=A[i];**

**}**

**}**

**else**

**{**

**printf("%d\n",A[i]);**

**last=A[i];**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.\*;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String str[] = br.readLine().split(" ");**

**Set<Integer> set = new HashSet<Integer>();**

**for (int a = 0; a < 10; a++) {**

**set.add(Integer.parseInt(str[a]));**

**}**

**List<Integer> list = new ArrayList<Integer>();**

**list.addAll(set);**

**Collections.sort(list);**

**for (int a = 0; a < list.size(); a++) {**

**System.out.println(list.get(a));**

**}**

**}**

**}**

ALGO-38**算法训练 接水问题**

时间限制：1.0s   内存限制：64.0MB

关键字：模拟

问题描述

　　学校里有一个水房，水房里一共装有m 个龙头可供同学们打开水，每个龙头每秒钟的 供水量相等，均为1。 现在有n 名同学准备接水，他们的初始接水顺序已经确定。将这些同学按接水顺序从1 到n 编号，i 号同学的接水量为wi。接水开始时，1 到m 号同学各占一个水龙头，并同时打 开水龙头接水。当其中某名同学j 完成其接水量要求wj 后，下一名排队等候接水的同学k 马上接替j 同学的位置开始接水。这个换人的过程是瞬间完成的，且没有任何水的浪费。即 j 同学第x 秒结束时完成接水，则k 同学第x+1 秒立刻开始接水。若当前接水人数n’不足m， 则只有n’个龙头供水，其它m−n’个龙头关闭。 现在给出n 名同学的接水量，按照上述接水规则，问所有同学都接完水需要多少秒。

输入格式

　　第1 行2 个整数n 和m，用一个空格隔开，分别表示接水人数和龙头个数。 第2 行n 个整数w1、w2、……、wn，每两个整数之间用一个空格隔开，wi 表示i 号同 学的接水量。

输出格式

　　输出只有一行，1 个整数，表示接水所需的总时间。

样例输入

5 3  
4 4 1 2 1

样例输出

4

样例输入

8 4  
23 71 87 32 70 93 80 76

样例输出

163

输入输出样例 1 说明

　　第1 秒，3 人接水。第1 秒结束时，1、2、3 号同学每人的已接水量为1，3 号同学接完  
　　水，4 号同学接替3 号同学开始接水。  
　　第2 秒，3 人接水。第2 秒结束时，1、2 号同学每人的已接水量为2，4 号同学的已接  
　　水量为1。  
　　第3 秒，3 人接水。第3 秒结束时，1、2 号同学每人的已接水量为3，4 号同学的已接  
　　水量为2。4 号同学接完水，5 号同学接替4 号同学开始接水。  
　　第4 秒，3 人接水。第4 秒结束时，1、2 号同学每人的已接水量为4，5 号同学的已接  
　　水量为1。1、2、5 号同学接完水，即所有人完成接水。  
　　总接水时间为4 秒。

数据规模和约定

　　1 ≤ n ≤ 10000，1 ≤m≤ 100 且m≤ n；  
　　1 ≤ wi ≤ 100。

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**#include "algorithm"**

**#include "stack"**

**#include "list"**

**#include "math.h"**

**using namespace std;**

**const int N =101;**

**struct X**

**{**

**int total;**

**int now;**

**X(int t=0,int n=0):total(t),now(n){}**

**}a[10001],b[201];**

**int main()**

**{**

**int n,m;**

**scanf("%d%d",&n,&m);**

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

**scanf("%d",&a[i].total);**

**int index=0;**

**for(int i=0;i<m;i++)**

**{**

**b[i]=a[ index++];**

**}**

**int time=0;**

**while(true)**

**{**

**time++;**

**for(int i=0;i<m;i++)**

**b[i].now++;**

**for(int i=0;i<m;i++)**

**{**

**if(b[i].now>=b[i].total)**

**{**

**if(index<n)**

**{**

**b[i]=a[index++];**

**}**

**}**

**}**

**int i;**

**for ( i=0;i<m;i++)**

**{**

**if(b[i].now<b[i].total)**

**break;**

**}**

**if(i==m)**

**break;**

**}**

**cout<<time<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <stdlib.h>**

**#include <malloc.h>**

**/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/**

**int max(int \*p,int n)**

**{**

**int Max = p[0];**

**int i;**

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

**{**

**if(p[i]>Max)**

**Max = p[i];**

**}**

**return Max;**

**}**

**int main(int argc, char \*argv[]) {**

**int m,n; //m个水龙头 n个同学**

**int i,j,second,min,num;**

**int \*p;**

**scanf("%d%d",&n,&m);**

**p = (int \*)malloc(n\*sizeof(int));**

**if( NULL == p )**

**{**

**printf("no enough memory!\n");**

**return 0;**

**}**

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&p[i]);**

**}**

**if( n<=m)**

**{**

**printf("%d",max(p,n));**

**}**

**if( n>m)**

**{**

**second = 0;**

**num = 0;**

**while(n>m)**

**{**

**min = p[0];**

**j = 0;**

**for(i=0;i<m;i++)**

**{**

**if(min>p[i])**

**{**

**min = p[i];**

**j=i;**

**}**

**}**

**for(i=0;i<m;i++)**

**{**

**p[i] -= min;**

**}**

**second += min;**

**p[j]=p[m+num]; //第m+num同学取代取完水的同学**

**num++;**

**n--;//取水同学减少一个**

**if( n<=m)**

**{**

**printf("%d",max(p,n)+second);**

**}**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**import java.util.Arrays;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(**

**new InputStreamReader(System.in));**

**String[] str1 = br.readLine().split(" ");**

**String[] str2 = br.readLine().split(" ");**

**int[] first = new int[2];**

**for (int i = 0; i < first.length; i++)**

**first[i] = Integer.parseInt(str1[i]);**

**int[] student = new int[first[0]];**

**for (int i = 0; i < student.length; i++) {**

**student[i] = Integer.parseInt(str2[i]);**

**}**

**if (first[0] > first[1])**

**System.out.println(getwater(student, first[1]));**

**else**

**System.out.println(getmax(student));**

**}**

**public static int getwater(int[] student, int watercome) {**

**int time = 0;**

**int[] firststudents = new int[watercome];**

**for (int i = 0; i < watercome; i++) {**

**firststudents[i] = student[i];**

**}**

**int k = firststudents.length;**

**for (int i = 0; i <= student.length - watercome; i++) {**

**if (i == student.length - watercome)**

**time += getmax(firststudents);**

**else {**

**int min = getmin(firststudents);**

**for (int j = 0; j < firststudents.length; j++) {**

**firststudents[j] -= min;**

**if (firststudents[j] == 0 && k<student.length) {**

**firststudents[j] = student[k++];**

**}**

**}**

**time += min;**

**}**

**}**

**return time;**

**}**

**public static int getmin(int[] arr) {**

**Arrays.sort(arr);**

**return arr[0];**

**}**

**public static int getmax(int[] arr) {**

**Arrays.sort(arr);**

**return arr[arr.length - 1];**

**}**

**}**

ALGO-37**算法训练 Hankson的趣味题**

时间限制：1.0s   内存限制：64.0MB

关键字：数论

问题描述

　　Hanks 博士是BT (Bio-Tech，生物技术) 领域的知名专家，他的儿子名叫Hankson。现 在，刚刚放学回家的Hankson 正在思考一个有趣的问题。 今天在课堂上，老师讲解了如何求两个正整数c1 和c2 的最大公约数和最小公倍数。现 在Hankson 认为自己已经熟练地掌握了这些知识，他开始思考一个“求公约数”和“求公 倍数”之类问题的“逆问题”，这个问题是这样的：已知正整数a0,a1,b0,b1，设某未知正整 数x 满足： 1． x 和a0 的最大公约数是a1； 2． x 和b0 的最小公倍数是b1。 Hankson 的“逆问题”就是求出满足条件的正整数x。但稍加思索之后，他发现这样的 x 并不唯一，甚至可能不存在。因此他转而开始考虑如何求解满足条件的x 的个数。请你帮 助他编程求解这个问题。

输入格式

　　输入第一行为一个正整数n，表示有n 组输入数据。  
  
　　接下来的n 行每 行一组输入数据，为四个正整数a0，a1，b0，b1，每两个整数之间用一个空格隔开。输入 数据保证a0 能被a1 整除，b1 能被b0 整除。

输出格式

　　输出共n 行。每组输入数据的输出结果占一行，为一个整数。  
　　对于每组数据：若不存在这样的 x，请输出0； 若存在这样的 x，请输出满足条件的x 的个数；

样例输入

2  
41 1 96 288  
95 1 37 1776

样例输出

6  
2

样例说明

　　第一组输入数据，x 可以是9、18、36、72、144、288，共有6 个。  
　　第二组输入数据，x 可以是48、1776，共有2 个。

数据规模和约定

　　对于 50%的数据，保证有1≤a0，a1，b0，b1≤10000 且n≤100。  
　　对于 100%的数据，保证有1≤a0，a1，b0，b1≤2,000,000,000 且n≤2000。

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<math.h>**

**const int maxn = 55000;**

**const int pmaxn = 5500;**

**int prime[pmaxn], p, ans = 1, tt;**

**bool vis[maxn];**

**int div(int & a, int b)**

**{**

**int cnt = 0;**

**while(a % b == 0)**

**{**

**a /= b;**

**cnt++;**

**}**

**return cnt;**

**}**

**int main()**

**{**

**int test = 50010;**

**int t = 0, n, m = (int)sqrt(maxn+0.5), i;**

**for(i = 2; i <= m; i++)**

**{**

**if(vis[i]) continue;**

**prime[t++] = i;**

**for(int j = i \* i; j < maxn; j += i)**

**vis[j] = true;**

**}**

**for(; i < maxn; i++)**

**if(!vis[i])**

**prime[t++] = i;**

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

**while(n--)**

**{**

**int a, aa, b, bb;**

**ans = 1;**

**tt = 0;**

**scanf("%d%d%d%d", &a, &aa, &b, &bb);**

**for(i = 0; bb != 1; i++)**

**{**

**if(i > 4950)**

**{**

**prime[i] = bb;**

**// break;**

**}**

**if(bb % prime[i] == 0)**

**{**

**int cnt = 100000000, type;**

**int a1c = div(aa, prime[i]);**

**int a0c = div(a, prime[i]);**

**if(a1c > a0c) continue;**

**if(a1c == a0c)**

**{**

**cnt = a1c;**

**type = 1;**

**}**

**if(a1c < a0c)**

**{**

**type = 2;**

**cnt = a1c;**

**}**

**int b0c = div(b, prime[i]);**

**int b1c = div(bb, prime[i]);**

**int tot;**

**if(b1c == b0c)**

**{**

**if(type == 1)**

**tot = b1c - cnt + 1;**

**else**

**{**

**if(cnt > b1c)**

**continue;**

**tot = 1;**

**}**

**if(tot <= 0) continue;**

**}**

**if(b1c > b0c)**

**{**

**if(type == 2)**

**{**

**if(b1c != cnt)**

**{**

**ans = 0;break;**

**}**

**tot = 1;**

**}**

**else**

**{**

**if(cnt > b1c)**

**continue;**

**tot = 1;**

**}**

**}**

**ans \*= tot;**

**}**

**}**

**printf("%d\n", ans);**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int gcd(int x,int y)**

**{**

**return y?gcd(y,x%y):x;**

**}**

**int main()**

**{**

**int n,a0,a1,b0,b1,x,cnt;**

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

**while(n--)**

**{**

**scanf("%d%d%d%d",&a0,&a1,&b0,&b1);**

**cnt=0;**

**for(x=1;x\*x<=b1;x++)**

**{**

**if(b1%x==0)//如果x确实是b1的约数**

**{**

**if(x%a1==0)//如果x确实是a1的倍数**

**{**

**if(gcd(x,a0)==a1&&(x/gcd(x,b0)\*b0==b1))//乘以b0放在后面，否则运算结果溢出，尤其是第2个输入用例**

**cnt++;**

**}**

**if((b1/x)%a1==0&&x\*x!=b1)//如果b1/x确实是a1的倍数,并且x与b1/x不相等**

**{**

**if(gcd(b1/x,a0)==a1&&(b1/x/gcd(b1/x,b0)\*b0==b1))//乘以b0放在后面，否则运算结果溢出**

**cnt++;**

**}**

**}**

**}**

**printf("%d\n",cnt);**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**import java.util.ArrayList;**

**public class Main {**

**private static ArrayList<Integer> arrayList = new ArrayList<Integer>();**

**private static Scanner scanner = new Scanner(System.in);**

**private static int[] prim = new int[20000];**

**private static int[] ans = new int[200000];**

**private static int[] num = new int[20000];**

**private static int end;**

**private static int len;**

**public static void main(String[] args) {**

**int n = scanner.nextInt();**

**while (n > 0) {**

**int a0 = scanner.nextInt();**

**int a1 = scanner.nextInt();**

**int b0 = scanner.nextInt();**

**int b1 = scanner.nextInt();**

**int t = len = end = 0;**

**work(b1, a0, a1);**

**for (int i = 0; i < len; i++)**

**if (gcd(ans[i], a0) == a1**

**&& b0 / gcd(b0, ans[i]) \* ans[i] == b1)**

**t++;**

**arrayList.add(t);**

**n--;**

**}**

**for (int i : arrayList)**

**System.out.println(i);**

**}**

**public static int gcd(int a, int b) {**

**int t;**

**while (b != 0) {**

**t = a % b;**

**a = b;**

**b = t;**

**}**

**return a;**

**}**

**public static void dfs(int now, int pro) {**

**if (now == end) {**

**ans[len++] = pro;**

**return;**

**}**

**dfs(now + 1, pro);**

**for (int i = 1; i <= num[now]; i++) {**

**pro \*= prim[now];**

**dfs(now + 1, pro);**

**}**

**}**

**public static void work(int n, int a0, int a1) {**

**int i = 2;**

**int lim = (int) Math.sqrt(n);**

**while (n != 1 && i <= lim) {**

**if (n % i == 0) {**

**if (a1 % i == 0 && a1 % i != 0) {**

**while (n % i == 0)**

**n /= i;**

**i++;**

**continue;**

**}**

**prim[end] = i;**

**num[end] = 0;**

**while (n % i == 0) {**

**num[end]++;**

**n /= i;**

**}**

**end++;**

**}**

**i++;**

**}**

**if (n != 1) {**

**prim[end] = n;**

**num[end++] = 1;**

**}**

**dfs(0, 1);**

**}**

**}**

ALGO-36**算法训练 传纸条**

时间限制：1.0s   内存限制：512.0MB

关键字：动态规划

问题描述

　　小渊和小轩是好朋友也是同班同学，他们在一起总有谈不完的话题。一次素质拓展活动中，班上同学安排做成一个m行n列的矩阵，而小渊和小轩被安排在矩阵对角线的两端，因此，他们就无法直接交谈了。幸运的是，他们可以通过传纸条来进行交流。纸条要经由许多同学传到对方手里，小渊坐在矩阵的左上角，坐标(1,1)，小轩坐在矩阵的右下角，坐标(m,n)。从小渊传到小轩的纸条只可以向下或者向右传递，从小轩传给小渊的纸条只可以向上或者向左传递。  
　　在活动进行中，小渊希望给小轩传递一张纸条，同时希望小轩给他回复。班里每个同学都可以帮他们传递，但只会帮他们一次，也就是说如果此人在小渊递给小轩纸条的时候帮忙，那么在小轩递给小渊的时候就不会再帮忙。反之亦然。  
　　还有一件事情需要注意，全班每个同学愿意帮忙的好感度有高有低（注意：小渊和小轩的好心程度没有定义，输入时用0表示），可以用一个0-100的自然数来表示，数越大表示越好心。小渊和小轩希望尽可能找好心程度高的同学来帮忙传纸条，即找到来回两条传递路径，使得这两条路径上同学的好心程度只和最大。现在，请你帮助小渊和小轩找到这样的两条路径。

输入格式

　　输入第一行有2个用空格隔开的整数m和n，表示班里有m行n列（1<=m,n<=50）。  
　　接下来的m行是一个m\*n的矩阵，矩阵中第i行j列的整数表示坐在第i行j列的学生的好心程度。每行的n个整数之间用空格隔开。

输出格式

　　输出一行，包含一个整数，表示来回两条路上参与传递纸条的学生的好心程度之和的最大值。

样例输入

3 3  
0 3 9  
2 8 5  
5 7 0

样例输出

34

数据规模和约定

　　30%的数据满足：1<=m,n<=10  
　　100%的数据满足：1<=m,n<=50

**参考代码：**

**C++**

**#include<iostream>**

**#include<string.h>**

**using namespace std ;**

**int dis[101][101] ;**

**int f[105][52][52] ;**

**int rows , cols , step ;**

**#define MAX(a,b) ((a) > (b) ? (a) : (b))**

**int Max( int a , int b , int c , int d){**

**a = MAX ( a , b ) ;**

**a = MAX ( a , c ) ;**

**a = MAX ( a , d ) ;**

**return a ;**

**}**

**int main(){**

**std::ios::sync\_with\_stdio(false) ;**

**int i , j , k ;**

**memset( f , 0 , sizeof(f) ) ;**

**memset( dis , 0 , sizeof(dis) ) ;**

**cin >> rows >> cols ;**

**for( i = 1 ; i <= rows ; i++ )**

**for( j = 1 ; j <= cols ; j++ )**

**cin >> dis[i][j] ;**

**step = rows + cols - 2 ;**

**f[1][2][1] = f[1][1][2]= dis[1][2] + dis[2][1] ;**

**for( k = 2 ; k <= (step - 1) ; k++ )**

**for( i = 1 ; i <= rows ; i++ )**

**for( j = 1 ; j <= rows ; j++ )**

**if( i != j )**

**f[k][i][j] = dis[i][k - i + 2] + dis[j][k - j + 2] + Max( f[k - 1][i - 1][j - 1] , f[k - 1][i - 1][j] , f[k - 1][i][j - 1] , f[k - 1][i][j]) ;**

**cout << f[step - 1][rows - 1][rows] << endl ;**

**return 0 ;**

**}**

**C**

**#include<stdio.h>**

**int n,m;**

**int i,j,k;**

**int Map[51][51];**

**int F[111][51][51];**

**int Max(int a,int b,int c,int d)**

**{**

**if(a>=b&&a>=c&&a>=d)**

**return a;**

**if(b>=a&&b>=c&&b>=d)**

**return b;**

**if(c>=a&&c>=b&&c>=d)**

**return c;**

**if(d>=a&&d>=b&&d>=c)**

**return d;**

**}**

**int main()**

**{**

**scanf("%d%d",&n,&m);**

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

**for(j=1;j<=m;j++)**

**scanf("%d",&Map[i][j]);**

**for(k=1;k<=n+m-2;k++)**

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

**for(j=1;j<=n;j++)**

**if(i==n&&j==n&&k==n+m-2)**

**F[k][i][j]=Max(F[k-1][i-1][j],F[k-1][i][j-1],F[k-1][i][j],F[k-1][i-1][j-1])+Map[i][k+2-i]+Map[j][k+2-j];**

**else if(i!=j&&k+2-i>=1&&k+2-j>=1)**

**F[k][i][j]=Max(F[k-1][i-1][j],F[k-1][i][j-1],F[k-1][i][j],F[k-1][i-1][j-1])+Map[i][k+2-i]+Map[j][k+2-j];**

**printf("%d",F[n+m-2][n][n]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**public static void main(String[] args) {**

**Scanner scan = new Scanner(System.in);**

**int m = scan.nextInt();**

**int n = scan.nextInt();**

**int[][] a = new int[m + 1][n + 1];**

**int k = m + n - 1;**

**int[][][] b = new int[k + 1][m + 1][m + 1];**

**for (int i = 1; i <= m; i++) {**

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

**a[i][j] = scan.nextInt();**

**}**

**}**

**for (int h = 2; h <= m + n - 1; h++) {**

**int g = h > m ? m : h;**

**for (int i = 1; i <= g; i++) {**

**for (int j = i; j <= g; j++) {**

**if (i == j && h < m + n - 1) {**

**continue;**

**}**

**int max = b[h - 1][i][j];**

**if (b[h - 1][i - 1][j] > max) {**

**max = b[h - 1][i - 1][j];**

**}**

**if (b[h - 1][i - 1][j - 1] > max) {**

**max = b[h - 1][i - 1][j - 1];**

**}**

**if (b[h - 1][i][j - 1] > max) {**

**max = b[h - 1][i][j - 1];**

**}**

**if (h - i + 1 <= n && h - j + 1 <= n)**

**b[h][i][j] = max + a[i][h - i + 1] + a[j][h - j + 1];**

**}**

**}**

**}**

**System.out.println(b[k][m][m]);**

**}**

**}**

ALGO-35**算法训练 传球游戏**

时间限制：1.0s   内存限制：256.0MB

关键字：动态规划

问题描述

　　上体育课的时候，小蛮的老师经常带着同学们一起做游戏。这次，老师带着同学们一起做传球游戏。  
　　游戏规则是这样的：n个同学站成一个圆圈，其中的一个同学手里拿着一个球，当老师吹哨子时开始传球，每个同学可以把球传给自己左右的两个同学中的一个（左右任意），当老师再次吹哨子时，传球停止，此时，拿着球没传出去的那个同学就是败者，要给大家表演一个节目。  
　　聪明的小蛮提出一个有趣的问题：有多少种不同的传球方法可以使得从小蛮手里开始传的球，传了m次以后，又回到小蛮手里。两种传球的方法被视作不同的方法，当且仅当这两种方法中，接到球的同学按接球顺序组成的序列是不同的。比如有3个同学1号、2号、3号，并假设小蛮为1号，球传了3次回到小蛮手里的方式有1->2->3->1和1->3->2->1，共2种。

输入格式

　　共一行，有两个用空格隔开的整数n，m（3<=n<=30，1<=m<=30）。

输出格式

　　t共一行，有一个整数，表示符合题意的方法数。

样例输入

3 3

样例输出

2

数据规模和约定

　　40%的数据满足：3<=n<=30，1<=m<=20  
　　100%的数据满足：3<=n<=30，1<=m<=30

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <iostream>**

**using namespace std;**

**int n,m;**

**int fun(int x)**

**{**

**if(x<1)**

**return x+n;**

**if(x>n) return x-n;**

**return x;**

**}**

**int main()**

**{ int f[31][31]={0},i,j;**

**scanf("%d%d",&n,&m);**

**f[1][2]=f[1][n]=1;**

**for (i=2;i<=m;i++)**

**for (j=1;j<=n;j++)**

**f[i][j]=f[i-1][fun(j-1)]+f[i-1][fun(j+1)];**

**printf("%d",f[m][1]);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int n,m;**

**int at1(int x)**

**{**

**if(x<1) return x + n;**

**if(x>n) return x - n;**

**return x;**

**}int main()**

**{**

**int f[31][31] = {0},i,j;**

**scanf("%d%d",&n,&m);**

**f[1][2] = f[1][n] = 1;**

**for(i = 2;i <= m;i++)**

**for(j = 1;j <= n;j++)**

**f[i][j] = f[i - 1][at1(j - 1)] + f[i - 1][at1(j + 1)];**

**printf("%d",f[m][1]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scan = new Scanner(System.in);**

**int n = scan.nextInt();**

**int m = scan.nextInt();**

**scan.close();**

**int[][] a = new int[m][n];**

**a[0][n - 1] = 1;**

**a[0][1] = 1;**

**for (int i = 1; i < m; i++) {**

**for (int j = 0; j < n; j++) {**

**a[i][j] = a[i - 1][(j - 1 + n) % n] + a[i - 1][(j + 1) % n];**

**}**

**}**

**System.out.println(a[m - 1][0]);**

**}**

**}**

ALGO-34**算法训练 纪念品分组**

时间限制：1.0s   内存限制：256.0MB

关键字：贪心 排序

问题描述

　　元旦快到了，校学生会让乐乐负责新年晚会的纪念品发放工作。为使得参加晚会的同学所获得的纪念品价值 相对均衡，他要把购来的纪念品根据价格进行分组，但每组最多只能包括两件纪念品，并且每组纪念品的价格之和不能超过一个给定的整数。为了保证在尽量短的时 间内发完所有纪念品，乐乐希望分组的数目最少。  
　　你的任务是写一个程序，找出所有分组方案中分组数最少的一种，输出最少的分组数目。

输入格式

　　输入包含n+2行：  
　　第1行包括一个整数w，为每组纪念品价格之和的上限。  
　　第2行为一个整数n，表示购来的纪念品的总件数。  
　　第3~n+2行每行包含一个正整数pi (5 <= pi <= w)，表示所对应纪念品的价格。

输出格式

　　输出仅一行，包含一个整数，即最少的分组数目。

样例输入

100  
9  
90  
20  
20  
30  
50  
60  
70  
80  
90

样例输出

6

数据规模和约定

　　50%的数据满足：1 <= n <= 15  
　　100%的数据满足：1 <= n <= 30000, 80 <= w <= 200

**参考代码：**

**C++**

**#include "iostream"**

**#include "string"**

**#include "stdio.h"**

**#include "ctype.h"**

**#include "algorithm"**

**#include "stack"**

**#include "list"**

**#include "math.h"**

**using namespace std;**

**const int N =30001;**

**int a[N];**

**int main()**

**{**

**int n;**

**int w;**

**cin>>w>>n;**

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

**{**

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

**}**

**std::sort(a,a+n);**

**int ans=0;**

**for(int i=0,j=n-1;i<=j;)**

**{**

**for(;i<j;j--)**

**{**

**if(a[i]+a[j]<=w)**

**break;**

**else ans++;**

**}**

**if(i!=j)**

**{**

**i++;j--;**

**ans++;**

**}**

**else**

**{**

**ans++;**

**break;**

**}**

**}**

**cout<<ans;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**void qsort(int i,int j);**

**int a[30000];**

**void qsort(int i,int j){**

**int x,p,q;**

**x=a[i]; p=i; q=j;**

**while (i<j) {**

**while ((i<j)&&(a[j]>x))**

**j--;**

**if (i<j)**

**{**

**a[i]=a[j];**

**i++;**

**}**

**while ((i<j)&&(a[i]<x))**

**i++;**

**if (i<j)**

**{**

**a[j]=a[i];**

**j--;**

**}**

**}**

**a[i]=x;**

**if (p<i-1)**

**qsort(p,i-1);**

**if (i+1<q)**

**qsort(i+1,q);}**

**main(){**

**int n,w,s,i,j;**

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

**for (i=0;i<n;i++)**

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

**qsort(0,n-1);**

**i=0;**

**j=n-1;**

**s=0;**

**while (i<j)**

**{**

**s++;**

**if (a[i]+a[j]<=w)**

**{**

**i++;**

**j--;**

**}**

**else j--;**

**}**

**if ((i==j)&&(a[i]<=w))**

**s++;**

**printf("%d",s);**

**getchar();**

**getchar();**

**return(0);**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.Arrays;**

**public class Main {**

**static int a[]=new int[30001];**

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

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

**int w=Integer.parseInt(bf.readLine());**

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

**for(int i=1;i<30001;i++)**

**a[i]=999999;**

**int k=n;**

**int sum=0;**

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

**a[i]=Integer.parseInt(bf.readLine());**

**Arrays.sort(a);**

**oter:for(int i=1;i<=n;i++){**

**for(int j=n;j>0;j--){**

**if(k==i){**

**sum++;**

**break oter;**

**}**

**if(k-i==1&&a[i]+a[k]>w){**

**sum+=2;**

**break oter;**

**}**

**if(k-i==1&&a[i]+a[k]<=w){**

**sum++;**

**break oter;**

**}**

**if(a[i]+a[k]>w){**

**k--;**

**sum++;**

**}**

**else{**

**sum++;**

**k--;**

**break;**

**}**

**}**

**}**

**System.out.println(sum);**

**}**

**}**

ALGO-33**算法训练 数列**

时间限制：1.0s   内存限制：256.0MB

关键字：数学 进制

问题描述

　　给定一个正整数k(3≤k≤15),把所有k的方幂及所有有限个互不相等的k的方幂之和构成一个递增的序列，例如，当k=3时，这个序列是：  
　　1，3，4，9，10，12，13，…  
　　（该序列实际上就是：30，31，30+31，32，30+32，31+32，30+31+32，…）  
　　请你求出这个序列的第N项的值（用10进制数表示）。  
　　例如，对于k=3，N=100，正确答案应该是981。

输入格式

　　只有1行，为2个正整数，用一个空格隔开：  
　　k N  
　　（k、N的含义与上述的问题描述一致，且3≤k≤15，10≤N≤1000）。

输出格式

　　计算结果，是一个正整数（在所有的测试数据中，结果均不超过2.1\*109）。（整数前不要有空格和其他符号）。

样例输入

3 100

样例输出

981

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cstdlib>**

**using namespace std;**

**int main()**

**{**

**int k, n;**

**cin>>k>>n;**

**int a[1010]={0,1,k,1+k,k\*k};**

**for (int i=5; i<=n; i++)**

**if (i%2) a[i]=a[i-1]+1; else a[i]=a[2]\*a[i/2];**

**cout<<a[n];**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int re[1000],k,n;**

**int p=0,t=1,i,j,f;**

**re[0]=1;j=0,f=1;**

**scanf("%d%d",&k,&n);**

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

**{**

**if(f==re[j])**

**{**

**f\*=k;**

**re[i]=f;**

**// printf("%d ",f);**

**j=0;**

**continue;**

**}**

**t=f+re[j++];**

**re[i]=t;**

**// printf("%d ",t);**

**}**

**printf("%d",re[i-1]);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main {**

**static long a[]=new long[1001];**

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

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

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

**int k=Integer.parseInt(s[0]);**

**int n=Integer.parseInt(s[1]);**

**int m=1;**

**int sum=0;**

**while(n>0){**

**a[m++]=n%2;**

**n=n/2;**

**}**

**m--;**

**for(int i=1;i<=m;i++){**

**sum+=a[i]\*add(k,i-1);**

**}**

**System.out.println(sum);**

**}**

**static int add(int k,int n){**

**int sum=1;**

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

**sum\*=k;**

**return sum;**

**}**

**}**

ALGO-32**算法训练 JAM计数法**

时间限制：1.0s   内存限制：256.0MB

关键字：组合生成

问题描述

　　Jam是个喜欢标新立异的科学怪人。他不使用阿拉伯数字计数，而是使用小写英文字母计数，他觉得这样做，会使世界更加丰富多彩。在他的计数法中，每个数字的位数都是相同的（使用相同个数的字母），英文字母按原先的顺序，排在前面的字母小于排在它后面的字母。我们把这样的“数字”称为Jam数字。在Jam数字中，每个字母互不相同，而且从左到右是严格递增的。每次，Jam还指定使用字母的范围，例如，从2到10，表示只能使用{b,c,d,e,f,g,h,i,j}这些字母。如果再规定位数为5，那么，紧接在Jam数字“bdfij”之后的数字应该是“bdghi”。（如果我们用U、V依次表示Jam数字“bdfij”与“bdghi”，则U<V< span>，且不存在Jam数字P，使U<P<V< span>）。你的任务是：对于从文件读入的一个Jam数字，按顺序输出紧接在后面的5个Jam数字，如果后面没有那么多Jam数字，那么有几个就输出几个。

输入格式

　　有2行，第1行为3个正整数，用一个空格隔开：  
　　s t w  
　　（其中s为所使用的最小的字母的序号，t为所使用的最大的字母的序号。w为数字的位数，这3个数满足：1≤s<T≤26, 2≤w≤t-s ）  
　　第2行为具有w个小写字母的字符串，为一个符合要求的Jam数字。  
　　所给的数据都是正确的，不必验证。

输出格式

　　最多为5行，为紧接在输入的Jam数字后面的5个Jam数字，如果后面没有那么多Jam数字，那么有几个就输出几个。每行只输出一个Jam数字，是由w个小写字母组成的字符串，不要有多余的空格。

样例输入

2 10 5  
bdfij

样例输出

bdghi  
bdghj  
bdgij  
bdhij  
befgh

**参考代码：**

**C++**

**#include<stdio.h>**

**int p[26], num;**

**void output()**

**{**

**for(int i = 0; i < num; i++)**

**printf("%c", p[i]+'a');**

**printf("\n");**

**}**

**int main()**

**{**

**int s, end;**

**scanf("%d%d%d", &s, &end, &num);**

**getchar();**

**for(int i = 0; i < num; i++)**

**{**

**char c = getchar();**

**p[i] = c - 'a';**

**}**

**int t = 5;**

**while(t--)**

**{**

**int crt = num - 1, tmp = end - 1;**

**while(crt >= 0 && p[crt--] == tmp--);**

**if(crt < 0)**

**return 0;**

**p[crt] = p[++crt]++;**

**while(++crt < num)**

**p[crt] = p[crt-1] + 1;**

**output();**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int s,t,w;**

**char in[26];**

**int ad[26],i,j,k,flag,st;**

**scanf("%d%d%d",&s,&t,&w);**

**getchar();**

**for(i=0;i<w;i++)**

**{**

**scanf("%c",&in[i]);**

**ad[i]=in[i]-'a'+1;**

**}**

**// for(i=0;i<w;i++)**

**// printf("%d ",ad[i]);//+'a'-1);**

**// printf("\n");**

**for(i=0;i<5;i++)**

**{**

**if(ad[0]==(t-w+1))**

**break;**

**// flag=0;**

**for(j=w-1;j>=0;j--)**

**{**

**if(ad[j]==t-w+j+1)**

**continue;**

**// if(!flag)**

**// {**

**// ad[j]++;**

**st=++ad[j];**

**// k=j;**

**for(k=j+1;k<w;k++)**

**ad[k]=++st;**

**for(k=0;k<w;k++)**

**printf("%c",ad[k]+'a'-1);**

**printf("\n");**

**flag=1;**

**break;**

**// }**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.util.\*;**

**public class Main {**

**public static void main(String args[]){**

**Scanner sc=new Scanner(System.in);**

**int s=sc.nextInt();**

**int t=sc.nextInt();**

**int w=sc.nextInt();**

**String str=sc.next();**

**oter: for(int i=0;i<5;i++){**

**for(int j=w-1;j>=0;j--){**

**if(str.charAt(j)-'a'+1==t-w+j+1){**

**if(j==0)**

**break oter;**

**continue;**

**}**

**else {**

**str=str.substring(0,j)+(char)(str.charAt(j)+1);**

**for(int k=j;k<w-1;k++)**

**str=str+(char)(str.charAt(k)+1);**

**break;**

**}**

**}**

**System.out.println(str);**

**}**

**}**

**}**

ALGO-31**算法训练 开心的金明**

时间限制：1.0s   内存限制：256.0MB

关键字：01背包 动态规划

问题描述

　　金明今天很开心，家里购置的新房就要领钥匙了，新房里有一间他自己专用的很宽敞的房间。更让他高兴的是，妈妈昨天对他说：“你的房间需要购买哪些物品，怎 么布置，你说了算，只要不超过N元钱就行”。今天一早金明就开始做预算，但是他想买的东西太多了，肯定会超过妈妈限定的N元。于是，他把每件物品规定了一 个重要度，分为5等：用整数1~5表示，第5等最重要。他还从因特网上查到了每件物品的价格（都是整数元）。他希望在不超过N元（可以等于N元）的前提 下，使每件物品的价格与重要度的乘积的总和最大。  
　　设第j件物品的价格为v[j]，重要度为w[j]，共选中了k件物品，编号依次为 j1，j2，……，jk，则所求的总和为：  
　　v[j1]\*w[j1]+v[j2]\*w[j2]+ …+v[jk]\*w[jk]。（其中\*为乘号）  
　　请 你帮助金明设计一个满足要求的购物单。

输入格式

　　输入文件 的第1行，为两个正整数，用一个空格隔开：  
　　N m  
　　（其中N（<30000）表示总钱 数，m（<25）为希望购买物品的个数。）  
　　从第2行到第m+1行，第j行给出了编号为j-1的物品的基本数据，每行有2个非负整数  
　　v p  
　　（其中v表示该物品的价格(v<=10000)，p表示该物品的重要度(1~5)）

输出格式

　　输出文件只有一个正整数，为不超过总钱数的物品的价格与重要度乘积的总和的最大值（<100000000）。

样例输入

1000 5  
800 2  
400 5  
300 5  
400 3  
200 2

样例输出

3900

数据规模和约定

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<string.h>**

**using namespace std;**

**int dp[30000];**

**int main()**

**{**

**int N,m,v[300],w[300];**

**//freopen("d://1.txt","r",stdin);**

**cin>>N>>m;**

**for(int i=0;i<m;i++)**

**{**

**cin>>v[i]>>w[i];**

**}**

**fill(dp,dp+N,0);**

**dp[0]=1;**

**for(int i=0;i<m;i++)**

**for(int j=N;j>=v[i];j--)**

**dp[j]=max(dp[j],dp[j-v[i]]+w[i]\*v[i]);**

**cout<<dp[N]<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**long mnum[30000];**

**long objprice[25]; //第i件商品的价格**

**long objk[25]; //第i件商品的价格与权值的积**

**int M,N; //M表示金钱总数，N表示商品总数**

**int main()**

**{**

**int i,m;**

**int maxp = 0;**

**scanf("%u%u", &M, &N); //M表示金钱总数，N表示商品总数**

**for(i=0;i<N;++i) //依次输入第i件商品的价格与权值，并计算出商品的价格与权值之积**

**{**

**scanf("%u%u", &objprice[i], &m);**

**objk[i] = m\*objprice[i];**

**}**

**for(i=0;i<N;++i)**

**{**

**for(m=0;m<M-objprice[i];++m)**

**if (mnum[m+objprice[i]] + objk[i] > mnum[m])**

**mnum[m] = mnum[m+objprice[i]] + objk[i];**

**}**

**for(i=0;i<M;++i)**

**if (mnum[i] > maxp)**

**maxp = mnum[i];**

**printf("%u", maxp);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String[] str = br.readLine().split(" ");**

**int money = Integer.parseInt(str[0]);**

**int count = Integer.parseInt(str[1]);**

**int[][] arr = new int[count][2];**

**for (int i = 0; i < count; i++) {**

**String[] order = br.readLine().split(" ");**

**for (int j = 0; j < 2; j++) {**

**arr[i][j] = Integer.parseInt(order[j]);**

**}**

**}**

**int[] temp = new int[money + 1];**

**for (int i = 0; i < count; i++) {**

**for (int j = money; j > 0; j--) {**

**if (j > arr[i][0]) {**

**if (temp[j - arr[i][0]] + arr[i][0] \* arr[i][1] > temp[j]) {**

**temp[j] = temp[j - arr[i][0]] + arr[i][0]**

**\* arr[i][1];**

**} else {**

**continue;**

**}**

**}**

**}**

**}**

**System.out.println(temp[money]);**

**}**

**}**

ALGO-30**算法训练 入学考试**

时间限制：1.0s   内存限制：256.0MB

关键字：0/1背包 动态规划

问题描述

　　辰辰是个天资聪颖的孩子，他的梦想是成为世界上最伟大的医师。为此，他想拜附近最有威望的医师为师。医师为了判断他的资质，给他出了一个难题。医师把他带到一个到处都是草药的山洞里对他说：“孩子，这个山洞里有一些不同的草药，采每一株都需要一些时间，每一株也有它自身的价值。我会给你一段时间，在这段时间里，你可以采到一些草药。如果你是一个聪明的孩子，你应该可以让采到的草药的总价值最大。”  
　　如果你是辰辰，你能完成这个任务吗？

输入格式

　　第一行有两个整数T（1 <= T <= 1000）和M（1 <= M <= 100），用一个空格隔开，T代表总共能够用来采药的时间，M代表山洞里的草药的数目。接下来的M行每行包括两个在1到100之间（包括1和100）的整数，分别表示采摘某株草药的时间和这株草药的价值。

输出格式

　　包括一行，这一行只包含一个整数，表示在规定的时间内，可以采到的草药的最大总价值。

样例输入

70 3  
71 100  
69 1  
1 2

样例输出

3

数据规模和约定

　　对于30%的数据，M <= 10；  
　　对于全部的数据，M <= 100。

**参考代码：**

**C++**

**#include<stdio.h>**

**const int maxn = 1010;**

**int dp[maxn];**

**int main()**

**{**

**int V, M, w[100], v[100], i, j;**

**scanf("%d%d", &V, &M);**

**for(i = 0; i < M; i++)**

**{**

**scanf("%d%d", &v[i], &w[i]);**

**}**

**for(i = 0; i < M; i++)**

**for(j = V; j >= v[i]; j--)**

**dp[j] = dp[j] > dp[j-v[i]] + w[i] ? dp[j] : dp[j-v[i]] + w[i];**

**printf("%d\n", dp[V]);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int totalTime, medics;**

**int value[100];**

**int time[100];**

**int max[1001];**

**int main()**

**{**

**scanf("%d%d", &totalTime, &medics);**

**int i, t;**

**for (i=0; i<medics; i++)**

**{**

**scanf("%d%d", &time[i], &value[i]);**

**}**

**for (i=0; i<medics; i++)**

**{**

**for (t=totalTime; t>0; t--)**

**{**

**if (time[i] <= t)**

**{**

**if (value[i] + max[t-time[i]] > max[t])//第i个的价值+不选第i个且用时为t-time[i-1]时最大价值**

**{**

**max[t] = value[i] + max[t-time[i]];**

**}**

**}**

**}**

**}**

**printf("%d\n", max[totalTime]);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**String[] str = br.readLine().split(" ");**

**int t = Integer.parseInt(str[0]);**

**int m = Integer.parseInt(str[1]);**

**int[][] arr = new int[m][2];**

**for (int i = 0; i < m; i++) {**

**String[] order = br.readLine().split(" ");**

**for (int j = 0; j < 2; j++) {**

**arr[i][j] = Integer.parseInt(order[j]);**

**}**

**}**

**int[] tag = new int[t + 1];**

**for (int i = 0; i < m; i++) {**

**for (int j = t; j > 0; j--) {**

**if (j >= arr[i][0]) {**

**if (tag[j - arr[i][0]] + arr[i][1] > tag[j]) {**

**tag[j] = tag[j - arr[i][0]] + arr[i][1];**

**}**

**} else {**

**continue;**

**}**

**}**

**}**

**System.out.println(tag[t]);**

**}**

**}**

ALGO-29**算法训练 校门外的树**

时间限制：1.0s   内存限制：256.0MB

关键字：区间处理

问题描述

　　某校大门外长度为L的马路上有一排树，每两棵相邻的树之间的间隔都是1米。我们可以把马路看成一个数轴，马路的一端在数轴0的位置，另一端在L的位置；数 轴上的每个整数点，即0，1，2，……，L，都种有一棵树。  
　　由于马路上有一些区域要用来建地铁。这些区域用它们在数轴上的起始点和终止点表示。已 知任一区域的起始点和终止点的坐标都是整数，区域之间可能有重合的部分。现在要把这些区域中的树（包括区域端点处的两棵树）移走。你的任务是计算将这些树 都移走后，马路上还有多少棵树。

输入格式

　　输入文件的第一行有两个整数L（1 <= L <= 10000）和 M（1 <= M <= 100），L代表马路的长度，M代表区域的数目，L和M之间用一个空格隔开。接下来的M行每行包含两个不同的整数，用一个空格隔开，表示一个区域的起始点 和终止点的坐标。

输出格式

　　输出文件包括一行，这一行只包含一个整数，表示马路上剩余的树的数目。

样例输入

500 3  
150 300  
100 200  
470 471

样例输出

298

数据规模和约定

　　对于20%的数据，区域之间没有重合的部分；  
　　对于其它的数据，区域之间有重合的情况。

**参考代码：**

**C++**

**#include<stdio.h>**

**#include<algorithm>**

**using namespace std;**

**struct Area**

**{**

**int l, r;**

**};**

**Area a[101];**

**bool cmp(Area a, Area b)**

**{**

**return a.l < b.l;**

**}**

**int main()**

**{**

**int L, m, i;**

**scanf("%d%d", &L, &m);**

**for(i = 0; i < m; i++)**

**scanf("%d%d", &a[i].l, &a[i].r);**

**sort(a, a+m, cmp);**

**int tot = 0, crt = -1;**

**for(i = 0; i < m; i++)**

**{**

**if(a[i].l > crt)**

**{**

**crt = a[i].l;**

**tot++;**

**}**

**if(crt < a[i].r)**

**{**

**tot += a[i].r - crt;**

**crt = a[i].r;**

**}**

**}**

**printf("%d\n", L-tot+1);**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**typedef struct**

**{**

**int start;**

**int end;**

**int flag;**

**}extent;**

**int main()**

**{**

**int L,M,i,j;**

**extent e[101];**

**scanf("%d%d",&L,&M);**

**for(i=1;i<=M;i++)**

**{**

**scanf("%d%d",&(e[i].start),&(e[i].end));**

**e[i].flag=1;**

**for(j=1;j<i;j++)**

**{**

**if(!(e[i].end<e[j].start||e[i].start>e[j].end)&&e[j].flag)**

**{**

**e[j].flag=0;**

**if(e[i].start>e[j].start)**

**e[i].start=e[j].start;**

**if(e[i].end<e[j].end)**

**e[i].end=e[j].end;**

**}**

**}//调整区间**

**}**

**for(i=1;i<=M;i++)**

**if(e[i].flag)**

**L=L-(e[i].end-e[i].start+1);**

**printf("%d",L+1);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in);**

**int m = input.nextInt();**

**int n = input.nextInt();**

**int[] a = new int[2 \* n];**

**int d = 0, e, max;**

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

**a[i] = input.nextInt();**

**a[i + n] = input.nextInt();**

**}**

**max = a[0];**

**for (int j = 0; j < 2 \* n; j++) {**

**if (max < a[j]) {**

**max = a[j];**

**}**

**}**

**int[] c = new int[max\*10];**

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

**for (int j = a[i]; j <=a[i + n]; j++) {**

**c[j] = 1;**

**}**

**}**

**for (int i = 0; i < max; i++) {**

**if (c[i] == 1) {**

**d = d + 1;**

**}**

**}**

**e = m - d;**

**System.out.println(e);**

**}**

**}**

ALGO-28**算法训练 星际交流**

时间限制：1.0s   内存限制：256.0MB

关键字：排列生成算法

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；

**参考代码：**

**C++**

**#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;**

**}**

**}**

**C**

**#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;**

**}**

**JAVA**

**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;**

**}**

**}**

ALGO-27**算法训练 FBI树**

时间限制：1.0s   内存限制：256.0MB

关键字：树 遍历

问题描述

　　我们可以把由“0”和“1”组成的字符串分为三类：全“0”串称为B串，全“1”串称为I串，既含“0”又含“1”的串则称为F串。  
　　FBI树是一种二叉树，它的结点类型也包括F结点，B结点和I结点三种。由一个长度为2N的“01”串S可以构造出一棵FBI树T，递归的构造方法如下：  
　　1)T的根结点为R，其类型与串S的类型相同；  
　　2)若串S的长度大于1，将串S从中间分开，分为等长的左右子串S1和S2；由左子串S1构造R的左子树T1，由右子串S2构造R的右子树T2。  
　　现在给定一个长度为2N的“01”串，请用上述构造方法构造出一棵FBI树，并输出它的后序遍历序列。

输入格式

　　第一行是一个整数N（0 <= N <= 10），第二行是一个长度为2N的“01”串。

输出格式

　　包括一行，这一行只包含一个字符串，即FBI树的后序遍历序列。

样例输入

3  
10001011

样例输出

IBFBBBFIBFIIIFF

数据规模和约定

　　对于40%的数据，N <= 2；  
　　对于全部的数据，N <= 10。  
　　注：  
　　[1] 二叉树：二叉树是结点的有限集合，这个集合或为空集，或由一个根结点和两棵不相交的二叉树组成。这两棵不相交的二叉树分别称为这个根结点的左子树和右子树。  
　　[2] 后序遍历：后序遍历是深度优先遍历二叉树的一种方法，它的递归定义是：先后序遍历左子树，再后序遍历右子树，最后访问根。

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstdlib>**

**#include<cstring>**

**int a[2100][3],f[1100],v=1,s=1;**

**void g(int x,int y)**

**{**

**if(y-x+1>1)**

**{**

**int vv=v,v1,v2;**

**a[vv][0]=v+1;v++;v1=v;**

**g(x,x+(y-x)/2);**

**a[vv][1]=v+1;v++;v2=v;**

**g(x+(y-x)/2+1,y);**

**if(a[v1][2]==a[v2][2])a[vv][2]=a[v1][2];**

**else a[vv][2]=2;**

**}**

**else{a[v][0]=-1;a[v][1]=-1;a[v][2]=f[s];s++;}**

**}**

**void h(int x)**

**{**

**if(a[x][0]!=-1)h(a[x][0]);**

**if(a[x][1]!=-1)h(a[x][1]);**

**if(a[x][2]==0)printf("B");**

**if(a[x][2]==1)printf("I");**

**if(a[x][2]==2)printf("F");**

**}**

**int main()**

**{**

**int n=1,i,nn;**

**char c;**

**scanf("%d\n",&nn);**

**for(i=1;i<=nn;i++)n\*=2;**

**for(i=1;i<=n;i++){scanf("%c",&c);f[i]=c-'0';}**

**g(1,n);**

**h(1);**

**printf("\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <malloc.h>**

**char in[1025];**

**typedef struct node**

**{**

**char data;**

**struct node \*lchild,\*rchild;**

**}Binode;**

**Binode \*create(int a,int b)**

**{**

**char ch;**

**Binode \*p;**

**int i,count0=0,count1=0,flag=0;**

**// scanf("%c",&ch);**

**// getchar();**

**if(a==b)**

**{**

**(in[a]=='0')?(ch='B'):(ch='I');**

**p=(Binode \*)malloc(sizeof(Binode));**

**p->data=ch;**

**p->lchild=NULL;**

**p->rchild=NULL;**

**return p;**

**}**

**for(i=a;i<=b;i++)**

**{**

**(in[i]=='0')?(count0++):(count1++);**

**if(count0 && count1)**

**{**

**ch='F';**

**flag=1;**

**break;**

**}**

**}**

**if(!flag)**

**{**

**if(!count0)**

**ch='I';**

**else**

**ch='B';**

**}**

**p=(Binode \*)malloc(sizeof(Binode));**

**p->data=ch;**

**p->lchild=create(a,((b+1)-a)/2+a-1);**

**p->rchild=create(((b+1)-a)/2+a,b);**

**return p;**

**}**

**void display(Binode \*p)**

**{**

**if(!p)**

**return;**

**display(p->lchild);**

**display(p->rchild);**

**printf("%c",p->data);**

**}**

**int main()**

**{**

**Binode \*r;**

**char c;**

**int p=0,n;**

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

**getchar();**

**while(1)**

**{**

**c=getchar();**

**if(c=='\n')**

**break;**

**in[p++]=c;**

**}**

**r=create(0,p-1);**

**display(r);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**static int N = 0;**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**N = sc.nextInt();**

**String s = sc.next();**

**nTree t = zuoTree(s);**

**System.out.println(t.HouXu());**

**}**

**static nTree zuoTree(String s) {**

**if (s.length() == 1) {**

**nTree t;**

**if (s.equals("1")) {**

**t = new nTree('I');**

**}else{**

**t = new nTree('B');**

**}**

**return t;**

**}**

**nTree t = new nTree();**

**int k = 0;**

**for(int i = 0;i<s.length();i++){**

**if(s.charAt(i) == '0'){**

**if(k == 0 || k == 1){**

**k = 1;**

**}**

**if(k == 2){**

**k = 3;**

**break;**

**}**

**}else{**

**if(k == 0 || k == 2){**

**k = 2;**

**}**

**if(k == 1){**

**k =3;**

**break;**

**}**

**}**

**}**

**if(k == 3){**

**t.value ='F';**

**}else if(k == 2){**

**t.value = 'I';**

**}else{**

**t.value = 'B';**

**}**

**t.left = zuoTree(s.substring(0,s.length()/2));**

**t.right = zuoTree(s.substring(s.length()/2));**

**return t;**

**}**

**}**

**class nTree {**

**nTree right;**

**nTree left;**

**char value;**

**nTree() {**

**right = null;**

**left = null;**

**value = ' ';**

**}**

**nTree(char v) {**

**right = null;**

**left = null;**

**value = v;**

**}**

**String HouXu() {**

**String s = "";**

**if (left != null) {**

**s = s + left.HouXu();**

**}**

**if (right != null) {**

**s = s + right.HouXu();**

**}**

**s = s + value;**

**return s;**

**}**

**}**

ALGO-26**算法训练 麦森数**

时间限制：1.0s   内存限制：256.0MB

关键字：二分 高精度

问题描述

　　形如2P-1的素数称为麦森数，这时P一定也是个素数。但反过来不一定，即如果P是个素数，2P-1不一定也是素数。到1998年底，人们已找到了37个麦森数。最大的一个是P=3021377，它有909526位。麦森数有许多重要应用，它与完全数密切相关。  
　　任务：从文件中输入P（1000<P<3100000），计算2P-1的位数和最后500位数字（用十进制高精度数表示）

输入格式

　　文件中只包含一个整数P（1000<P<3100000）

输出格式

　　第一行：十进制高精度数2P-1的位数。  
　　第2-11行：十进制高精度数2P-1的最后500位数字。（每行输出50位，共输出10行，不足500位时高位补0）  
　　不必验证2P-1与P是否为素数。

样例输入

1279

样例输出

386  
00000000000000000000000000000000000000000000000000  
00000000000000000000000000000000000000000000000000  
00000000000000104079321946643990819252403273640855  
38615262247266704805319112350403608059673360298012  
23944173232418484242161395428100779138356624832346  
49081399066056773207629241295093892203457731833496  
61583550472959420547689811211693677147548478866962  
50138443826029173234888531116082853841658502825560  
46662248318909188018470682222031405210266984354887  
32958028878050869736186900714720710555703168729087

**参考代码：**

**C++**

**#include<iostream>**

**#include<vector>**

**#include<cmath>**

**#include<iomanip>**

**using std::cin;**

**using std::cout;**

**using std::endl;**

**using std::vector;**

**using std::setw;**

**using std::setfill;**

**vector<int> Multiply(vector<int> ivec1,vector<int> ivec2)**

**{**

**vector<int> ivec(125,0);**

**for(int i=0;i!=125;++i)**

**{**

**int n=0;**

**for(int j=0;j!=125-i;++j)**

**{**

**int tmp=ivec[i+j]+ivec1[j]\*ivec2[i]+n;**

**ivec[i+j]=tmp%10000;**

**n=tmp/10000;**

**}**

**}**

**return ivec;**

**}**

**int main()**

**{**

**int p=0;**

**cin>>p;**

**cout<<(int)(p\*log10(2.0))+1<<endl;**

**vector<int> Boolvec;**

**while(p)**

**{**

**Boolvec.push\_back(p%2);**

**p/=2;**

**}**

**vector<int> ivec;**

**vector<int> Result;**

**ivec.push\_back(2);**

**Result.push\_back(1);**

**for(int i=1;i!=125;++i)**

**{**

**ivec.push\_back(0);**

**Result.push\_back(0);**

**}**

**for(int i=0;i<(int)Boolvec.size();++i)**

**{**

**if(Boolvec[i])**

**Result=Multiply(Result,ivec);**

**ivec=Multiply(ivec,ivec);**

**}**

**Result[0]--;**

**for(int i=124;i>=0;--i)**

**{**

**if(i%25==12)**

**{**

**cout<<setfill('0')<<setw(2)<<Result[i]/100;**

**cout<<endl;**

**cout<<setfill('0')<<setw(2)<<Result[i]%100;**

**}**

**else**

**{**

**if(i%25==0)**

**{**

**cout<<setfill('0')<<setw(4)<<Result[i];**

**cout<<endl;**

**}**

**else**

**cout<<setfill('0')<<setw(4)<<Result[i];**

**}**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**#include <math.h>**

**#define LEN 125 //每数组元素存放十进制的4位，因此数组最多只要125个元素即可**

**//Multiply函数功能是计算高精度乘法a\*b，结果的末500位放在a中**

**void Multiply(int \*a,int \*b)**

**{**

**int i,j;**

**int nCarry; //存放进位**

**int nTmp;**

**int c[LEN]; //存放结果的末500位**

**memset(c,0,sizeof(int)\*LEN);**

**for(i=0;i<LEN;i++)**

**{**

**nCarry=0;**

**for(j=0;j<LEN-i;j++)**

**{**

**nTmp=c[i+j]+a[j]\*b[i]+nCarry;**

**c[i+j]=nTmp%10000;**

**nCarry=nTmp/10000;**

**}**

**}**

**memcpy(a,c,LEN\*sizeof(int));**

**}**

**int main()**

**{**

**int i,p;**

**int anPow[LEN]; //存放不断增长的2的次幂**

**int aResult[LEN]; //存放最终结果的末500位**

**scanf("%d",&p);**

**printf("%d\n",(int)(p\*log10(2))+1);**

**//下面将2的次幂初始化为2^(2^0)(a^b表示a的b次方),最终结果初始化为1**

**anPow[0]=2;**

**aResult[0]=1;**

**for(i=1;i<LEN;i++)**

**{**

**anPow[i]=0;**

**aResult[i]=0;**

**}**

**//下面计算2的p次方**

**while(p>0) //p=0则说明p中的有效位都用过了，不需要再计算下去**

**{**

**if(p&1) //判断此时p中最低位是否为1**

**Multiply(aResult,anPow);**

**p>>=1;**

**Multiply(anPow,anPow);**

**}**

**aResult[0]--; //2的p次方算出后减1**

**//输出结果**

**for(i=LEN-1;i>=0;i--)**

**{**

**if(i%25==12)**

**printf("%02d\n%02d",aResult[i]/100,aResult[i]%100);**

**else**

**{**

**printf("%04d",aResult[i]);**

**if(i%25==0)**

**printf("\n");**

**}**

**}**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.math.BigInteger;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**int p = Integer.parseInt(br.readLine());**

**String result;**

**BigInteger a = new BigInteger("2");**

**result = pow(a, p).toString();**

**int count = result.length();**

**for (int b = 0; b < 500 - count; b++)**

**result = 0 + result;**

**System.out.println((int) (p \* Math.log10(2)) + 1);**

**for (int b = 0; b < 10; b++)**

**System.out.println(result.substring(b \* 50, (b + 1) \* 50));**

**}**

**public static BigInteger pow(BigInteger a, int b) {**

**BigInteger x = new BigInteger("1");**

**String s;**

**int len;**

**while (b > 0) {**

**if (b % 2 == 1) {**

**x = x.multiply(a);**

**s = x.toString();**

**len = s.length();**

**if (len > 500) {**

**x = new BigInteger(s.substring(len - 500, len));**

**}**

**}**

**a = a.pow(2);**

**s = a.toString();**

**len = s.length();**

**if (len > 500) {**

**a = new BigInteger(s.substring(len - 500, len));**

**}**

**b /= 2;**

**}**

**return x.subtract(new BigInteger("1"));**

**}**

**}**

ALGO-25**算法训练 Car的旅行路线**

时间限制：1.0s   内存限制：256.0MB

关键字：最短路

问题描述

　　又到暑假了，住在城市A的Car想和朋友一起去城市B旅游。她知道每个城市都有四个飞机场，分别位于一个矩形的四个顶点上，同一个城市中两个机场之间有一 条笔直的高速铁路，第I个城市中高速铁路了的单位里程价格为Ti，任意两个不同城市的机场之间均有航线，所有航线单位里程的价格均为t。  
　　那么Car应如何安排到城市B的路线才能尽可能的节省花费呢?她发现这并不是一个简单的问题，于是她来向你请教。  
　　找出一条从城市A到B的旅游路线，出发和到达城市中的机场可以任意选取，要求总的花费最少。

输入格式

　　的第一行有四个正整数s，t，A，B。  
　　S表示城市的个数，t表示飞机单位里程的价格，A，B分别为城市A，B的序号，(1<=A，B<=S)。  
　　接下来有S行，其中第I行均有7个正整数xi1，yi1，xi2，yi2，xi3，yi3，Ti，这当中的(xi1，yi1)，(xi2，yi2)，(xi3，yi3)分别是第I个城市中任意三个机场的坐标，T I为第I个城市高速铁路单位里程的价格。

输出格式

　　共有n行，每行一个数据对应测试数据，保留一位小数。

样例输入

1  
1 10 1 3  
1 1 1 3 3 1 30  
2 5 7 4 5 2 1  
8 6 8 8 11 6 3

样例输出

47.55

数据规模和约定

　　0<S<=100，

**参考代码：**

**C++**

**#include <float.h>**

**#include <math.h>**

**#include <stdio.h>**

**#include<cstring>**

**#define Sqr(x) ((x)\*(x))**

**double s[400][400], x[400], y[400];**

**int main()**

**{**

**int a, b, n, tt,t,i,j,k,N;**

**double res = DBL\_MAX;**

**scanf("%d%d%d%d", &n, &tt, &a, &b);**

**a-=1;**

**b-=1;**

**for ( i = 0; i < n; ++i)**

**{**

**for ( j = 0; j < 3; ++j)**

**scanf("%lf%lf", &x[i\*4+j], &y[i\*4+j]);**

**scanf("%d", &t);**

**for (j = 0; j < 3; ++j)**

**{**

**double x1 = x[i\*4+(j+1)%3], y1 = y[i\*4+(j+1)%3],**

**x2 = x[i\*4+(j+2)%3], y2 = y[i\*4+(j+2)%3];**

**if (fabs((x[i\*4+j]-x1)\*(x[i\*4+j]-x2)+(y[i\*4+j]-y1)\*(y[i\*4+j]-y2)) < 1E-7)**

**x[i\*4+3] = x1+x2-x[i\*4+j], y[i\*4+3] = y1+y2-y[i\*4+j];**

**}**

**for ( j = 0; j < 4; ++j)**

**for (k = 0; k < 4; ++k)**

**s[i\*4+j][i\*4+k] = t\*sqrt(Sqr(x[i\*4+j]-x[i\*4+k])+Sqr(y[i\*4+j]-y[i\*4+k]));**

**for ( j = 0; j < i\*4; ++j)**

**for ( k = 0; k < 4; ++k)**

**s[i\*4+k][j] = tt\*sqrt(Sqr(x[i\*4+k]-x[j])+Sqr(y[i\*4+k]-y[j])),**

**s[j][i\*4+k] = s[i\*4+k][j];**

**}**

**for ( k = 0; k < n\*4; ++k)**

**for (i = 0; i < n\*4; ++i)**

**for ( j = 0; j < n\*4; ++j)**

**if (s[i][k]+s[k][j] < s[i][j])**

**s[i][j] = s[i][k]+s[k][j];**

**for (i = 0; i < 4; ++i)**

**for ( j = 0; j < 4; ++j)**

**if (s[a\*4+i][b\*4+j] < res)**

**res = s[a\*4+i][b\*4+j];**

**printf("%.1lf\n", res);**

**return 0;**

**}**

**C**

**JAVA**

ALGO-24**算法训练 统计单词个数**

时间限制：1.0s   内存限制：256.0MB

关键字：字符串

问题描述

　　给出一个长度不超过200的由小写英文字母组成的字母串(约定;该字串以每行20个字母的方式输入，且保证每行一定为20个)。要求将此字母串分成k份 (1<k<=40)，且每份中包含的单词个数加起来总数最大(每份中包含的单词可以部分重叠。当选用一个单词之后，其第一个字母不能再用。例 如字符串this中可包含this和is，选用this之后就不能包含th)。  
　　单词在给出的一个不超过6个单词的字典中。  
　　要求输出最大的个数。

输入格式

　　第一行有二个正整数(p，k)  
　　p表示字串的行数;  
　　k表示分为k个部分。  
　　接下来的p行，每行均有20个字符。  
　　再接下来有一个正整数s，表示字典中单词个数。(1<=s<=6)  
　　接下来的s行，每行均有一个单词。

输出格式

　　每行一个整数，分别对应每组测试数据的相应结果。

样例输入

1 3  
thisisabookyouareaoh  
4  
is  
a  
ok  
sab

样例输出

7

数据规模和约定

　　长度不超过200，1<k<=40，字典中的单词数不超过6。

**参考代码：**

**C++**

**#include <stdio.h>**

**#include <string.h>**

**int N;**

**char s[201],Word[6][10];**

**int Find(int a,int b)**

**{**

**int i,j,k,l,t;**

**for (t=0,i=a;i<=b;i++)**

**for (j=0;j<N;j++)**

**{**

**for (k=i,l=0;k<=b&&Word[j][l]&&s[k]==Word[j][l];k++,l++);**

**if (Word[j][l]==0)**

**{**

**t++;**

**break;**

**}**

**}**

**return t;**

**}**

**int main()**

**{**

**int a[200][200],f[200];**

**int i,j,k,K,L;**

**scanf("%d%d",&N,&K);**

**for (L=0;N>0;N--)**

**{**

**getchar();**

**for (j=0;j<20;j++)**

**scanf("%c",&s[L++]);**

**}**

**s[L]=0;**

**scanf("%d",&N);**

**for (i=0;i<N;i++)**

**scanf("%s",Word[i]);**

**for (i=0;i<L;f[i]=a[0][i],i++)**

**for (j=i;j<L;j++)**

**a[i][j]=Find(i,j);**

**for (i=1;i<K;i++)**

**for (j=L-1;j>=i;j--)**

**for (f[j]=0,k=i;k<j;k++)**

**if (f[k]+a[k+1][j]>f[j])**

**f[j]=f[k]+a[k+1][j];**

**printf("%d",f[L-1]);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**#define InfiniteMin -999999999**

**int p,k,s;**

**char str[10][21];**

**char word[6][16];**

**int flag[6];**

**int Cnt[201][201];**

**int getCnt(int a,int b)**

**{**

**int i,j,m,count=0;**

**if(Cnt[a][b]<=0)**

**{**

**for (i=a;i<=b;i++)**

**for (j=0;j<s;j++)**

**if(word[j][0]==str[i/20][i%20])**

**{**

**for(m=1;word[j][m]!='\0'&&i+m<=b;m++)**

**if(word[j][m]!=str[(i+m)/20][(i+m)%20])**

**break;**

**if(word[j][m]=='\0')**

**{**

**count++;**

**break;**

**}**

**}**

**Cnt[a][b]=count;**

**}**

**return Cnt[a][b];**

**}**

**int main()**

**{**

**int i,u,K;**

**int max,temp;**

**int F[201][41];**

**scanf("%d%d",&p,&k);**

**for(i=0;i<p;i++)**

**scanf("%s",str[i]);**

**scanf("%d",&s);**

**for(i=0;i<s;i++)**

**scanf("%s",word[i]);**

**for(i=0;i<20\*p;i++)**

**F[i][1]=getCnt(0,i);**

**for(K=2;K<=k;K++)**

**for(i=k;i<p\*20;i++)**

**{**

**max=InfiniteMin;**

**for(u=K-1;u<=i-1;u++)**

**{**

**temp=F[u][K-1]+getCnt(u+1,i);**

**max=max>temp?max:temp;**

**}**

**F[i][K]=max;**

**}**

**printf("%d",F[p\*20-1][k]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main{**

**static int p;**

**static int k;**

**static String S;**

**static int wordsize;**

**static String[] Word;**

**static int num = 0;**

**static int maxnum = 0;**

**static int kefenge = 0;**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**p = sc.nextInt();**

**k = sc.nextInt();**

**S = "";**

**sc.nextLine();**

**for (int i = 0; i < p; i++) {**

**S = S + sc.nextLine();**

**}**

**wordsize = sc.nextInt();**

**Word = new String[wordsize];**

**for (int i = 0; i < wordsize; i++) {**

**Word[i] = sc.next();**

**}**

**zhaodanci(S);**

**if(kefenge<k){**

**num = num-(k-1);**

**}**

**System.out.println(num);**

**// System.out.println(kefenge);**

**}**

**public static void zhaodanci(String s){**

**boolean meizhaodao = true;**

**for(int i = 0;i<s.length();i++){**

**for(int j = 0;j<wordsize;j++){**

**if(s.substring(i,s.length()).length()<Word[j].length()){**

**continue;**

**}**

**if(s.substring(i,i+Word[j].length()).length()<Word[j].length()){**

**continue;**

**}**

**if(s.substring(i,i+Word[j].length()).equals(Word[j])){**

**// System.out.println(s+":"+Word[j]);**

**num++;**

**if(i != 0){**

**kefenge++;**

**}**

**zhaodanci(s.substring(i+1,s.length()));**

**return;**

**}**

**}**

**}**

**}**

**}**

ALGO-23**算法训练 一元三次方程求解**

时间限制：1.0s   内存限制：256.0MB

关键字：解方程

问题描述

　　有形如：ax3+bx2+cx+d=0 这样的一个一元三次方程。给出该方程中各项的系数(a，b，c，d 均为实数)，并约定该方程存在三个不同实根(根的范围在-100至100之间)，且根与根之差的绝对值>=1。要求三个实根。。

输入格式

　　四个实数：a，b，c，d

输出格式

　　由小到大依次在同一行输出这三个实根(根与根之间留有空格)，并精确到小数点后2位

样例输入

1 -5 -4 20

样例输出

-2.00 2.00 5.00

数据规模和约定

　　|a|，|b|，|c|，|d|<=10

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cmath>**

**using namespace std;**

**int main()**

**{**

**double a, b, c, d;**

**scanf("%lf%lf%lf%lf",&a,&b,&c,&d);**

**for (double x= -100.00; x <= 100.00; x+= 0.001)**

**{**

**if (fabs(a\*x\*x\*x + b\*x\*x + c\*x + d) <= 0.01)**

**{printf("%.2lf ",x); x+=1;}**

**}**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**float a,b,c,d,x=-100,t,y;**

**float f(float z)**

**{**

**return a\*z\*z\*z+b\*z\*z+c\*z+d;**

**}**

**int main()**

**{**

**scanf("%f%f%f%f",&a,&b,&c,&d);**

**t=f(x);**

**while(x<=100)**

**{**

**y=f(x);**

**if(y\*t<=0)printf("%.2f ",x);**

**x=x+0.001;t=y;**

**}**

**printf("\n");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**// TODO Auto-generated method stub**

**Scanner sc=new Scanner(System.in);**

**double f[]=new double[1000];**

**int t=0;**

**double a=sc.nextDouble(),b=sc.nextDouble(),c=sc.nextDouble(),d=sc.nextDouble();**

**for(double i=-100;i<=100;i+=0.001)**

**if(Math.abs(a\*i\*i\*i+b\*i\*i+c\*i+d)<0.0001)**

**{**

**System.out.printf("%.2f",i);System.out.print(" ");**

**i+=0.999;**

**}**

**}**

**}**

ALGO-22**算法训练 数的划分**

时间限制：1.0s   内存限制：256.0MB

关键字：动态规划

问题描述

　　将整数n分成k份，且每份不能为空，任意两份不能相同(不考虑顺序)。  
　　例如：n=7，k=3，下面三种分法被认为是相同的。  
　　1，1，5; 1，5，1; 5，1，1;  
　　问有多少种不同的分法。

输入格式

　　n，k

输出格式

　　一个整数，即不同的分法

样例输入

7 3

样例输出

4 {四种分法为：1，1，5;1，2，4;1，3，3;2，2，3;}

数据规模和约定

　　6<n<=200，2<=k<=6

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstring>**

**using namespace std;**

**int main()**

**{**

**int n,k,f[7][201];**

**memset(f,0,sizeof(f));**

**cin>>n>>k;**

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

**f[1][i]=1;**

**for(int i=2;i<=k;i++)**

**{**

**for(int j=0;j<=n-k;j++)**

**{**

**if(j>=i)**

**f[i][j]=f[i-1][j]+f[i][j-i];**

**else**

**f[i][j]=f[i-1][j];**

**}**

**}**

**cout<<f[k][n-k]<<endl;**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**int main()**

**{**

**int i,j,n,k,a[201][7]={0};**

**a[1][1] = 1;**

**scanf("%d%d",&n,&k);**

**for (i = 2; i <= n; i++)**

**for (j = 1; j <= k; j++)**

**if (i >= j)**

**a[i][j] = a[i - j][j] + a[i - 1][j - 1];**

**printf("%d",a[n][k]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner s = new Scanner(System.in);**

**int n = s.nextInt();**

**int m = s.nextInt();**

**int f[][] = new int[250][10];**

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

**f[i][1] = 1;**

**}**

**for (int i = 2; i <= m; i++) {**

**for (int j = 0; j <= n - m; j++) {**

**if (i > j) {**

**f[j][i] = f[j][i - 1];**

**} else {**

**f[j][i] = f[j][i - 1] + f[j - i][i];**

**}**

**}**

**}**

**System.out.println(f[n - m][m]);**

**}**

**}**

ALGO-21**算法训练 装箱问题**

时间限制：1.0s   内存限制：256.0MB

关键字：动态规划

问题描述

　　有一个箱子容量为V（正整数，0＜＝V＜＝20000），同时有n个物品（0＜n＜＝30），每个物品有一个体积（正整数）。  
　　要求n个物品中，任取若干个装入箱内，使箱子的剩余空间为最小。

输入格式

　　第一行为一个整数，表示箱子容量；  
　　第二行为一个整数，表示有n个物品；  
　　接下来n行，每行一个整数表示这n个物品的各自体积。

输出格式

　　一个整数，表示箱子剩余空间。

样例输入

24  
6  
8  
3  
12  
7  
9  
7

样例输出

0

**参考代码：**

**C++**

**#include<iostream>**

**#include<cstdio>**

**#include<cstdlib>**

**#include<cstring>**

**using namespace std;**

**bool v[21000];**

**int main()**

**{**

**//memset(v,0,sizeof(v));**

**int i,j,m,n,a;**

**scanf("%d%d",&m,&n);**

**v[0]=1;**

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

**{**

**scanf("%d",&a);**

**for(j=m;j>=a;j--)**

**v[j]=v[j]||v[j-a];**

**}**

**for(j=m;j>=0;j--)**

**{**

**if(v[j])**

**{**

**printf("%d\n",m-j);**

**return 0;**

**}**

**}**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**#include <malloc.h>**

**#define V 20001**

**#define N 30**

**int main()**

**{**

**int f[V];**

**int i,j;**

**int n,v;**

**int T[V];**

**memset(f,0,sizeof(f));**

**memset(T,0,sizeof(T));**

**scanf("%d%d",&n,&v);**

**for(i=1;i<=v;i++)**

**scanf("%d",&f[i]);**

**for(i=1;i<=v;i++)**

**{**

**for(j=n;j>=f[i];j--)**

**{**

**if(f[i]<=j&&f[i]+T[j-f[i]]>T[j])**

**T[j] = f[i]+T[j-f[i]];**

**}**

**}**

**printf("%d\n",n-T[n]);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.\*;**

**public class Main {**

**private static int v[][]=new int[31][20001];**

**public static void main (String args[])throws Exception{**

**//Scanner sc=new Scanner(System.in);**

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

**//int n=sc.nextInt();**

**//int m=sc.nextInt();**

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

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

**int a[]=new int[m+1];**

**for(int i=1;i<=m;i++)**

**a[i]=Integer.parseInt(bf.readLine());**

**//a[i]=sc.nextInt();**

**for(int i=1;i<=m;i++){**

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

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

**v[i][j]=v[i-1][j];**

**else**

**v[i][j]=Math.max(v[i-1][j], v[i-1][j-a[i]]+a[i]);**

**}**

**}**

**System.out.println(n-v[m][n]);**

**}**

**}**

ALGO-20**算法训练 求先序排列**

时间限制：1.0s   内存限制：256.0MB

关键字：递归

问题描述

　　给出一棵二叉树的中序与后序排列。求出它的先序排列。（约定树结点用不同的大写字母表示，长度<=8）。

输入格式

　　两行，每行一个字符串，分别表示中序和后序排列

输出格式

　　一个字符串，表示所求先序排列

样例输入

BADC  
BDCA

样例输出

ABCD

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstring>**

**char m[10000];**

**char h[10000];**

**void dfs(int ai,int aj,int bi,int bj)**

**{**

**int i;**

**{**

**if(aj<ai||bj<bi)**

**return ;**

**}**

**if(ai==aj)**

**printf("%c",m[ai]);**

**else**

**{**

**printf("%c",h[bj]);**

**for(i=ai;i<=aj;i++)**

**{**

**if(m[i]==h[bj])**

**{**

**break;**

**}**

**}**

**dfs(ai,i-1,bi,bi+(i-ai-1));**

**dfs(i+1,aj,bi+i-ai,bj-1);**

**}**

**}**

**int main()**

**{**

**gets(m);**

**gets(h);**

**dfs(0,strlen(m)-1,0,strlen(m)-1);**

**puts("");**

**return 0;**

**}**

**C**

**#include"stdio.h"**

**#include"string.h"**

**void dg(char z[],char h[])**

**{**

**if(strlen(h)==0)**

**return;**

**printf("%c",h[strlen(h)-1]);**

**if(strlen(h)==1)**

**return;**

**if(strlen(h)==2)**

**{**

**printf("%c",h[0]);**

**return;**

**}**

**char a[9],b[9];**

**int i,j;**

**for(i=0;z[i]!=h[strlen(h)-1];i++)**

**a[i]=z[i],b[i]=h[i];**

**a[i]='\0';**

**b[i++]='\0';**

**dg(a,b);**

**for(j=0;i<strlen(h);i++,j++)**

**a[j]=z[i],b[j]=h[i-1];**

**a[j]='\0';**

**b[j]='\0';**

**dg(a,b);**

**}**

**int main()**

**{**

**char h[9],z[9];**

**scanf("%s",z);**

**scanf("%s",h);**

**dg(z,h);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**while (scanner.hasNext()) {**

**String str1 = scanner.nextLine();**

**String str2 = scanner.nextLine();**

**showResult(str1, str2);**

**System.out.println();**

**}**

**}**

**private static void showResult(String str1, String str2) {**

**char chl = str2.charAt(str2.length() - 1);**

**System.out.print(chl);**

**int index = str1.indexOf(chl);**

**if (index > 0) {**

**showResult(str1.substring(0, index), str2.substring(0, index));**

**}**

**if (index < str1.length() - 1) {**

**showResult(str1.substring(index + 1),**

**str2.substring(index, str2.length() - 1));**

**}**

**}**

**}**

ALGO-19**算法训练 方格取数**

时间限制：1.0s   内存限制：256.0MB

关键字：动态规划

问题描述

　　设有N\*N的方格图(N<=10),我们将其中的某些方格中填入正整数,而其他的方格中则放入数字0。  
　　某人从图的左上角的A 点(1,1)出发，可以向下行走，也可以向右走，直到到达右下角的B点(N,N)。在走过的路上，他可以取走方格中的数（取走后的方格中将变为数字0）。  
　　此人从A点到B 点共走两次，试找出2条这样的路径，使得取得的数之和为最大。

输入格式

　　输入的第一行为一个整数N（表示N\*N的方格图），接下来的每行有三个整数，前两个表示位置，第三个数为该位置上所放的数。一行单独的0表示输入结束。

输出格式

　　只需输出一个整数，表示2条路径上取得的最大的和。

样例输入

8  
2 3 13  
2 6 6  
3 5 7  
4 4 14  
5 2 21  
5 6 4  
6 3 15  
7 2 14  
0 0 0

样例输出

67

**参考代码：**

**C++**

**#include<stdio.h>**

**const int maxn = 10;**

**int map[maxn][maxn], dp[maxn][maxn][maxn][maxn];**

**#define max(a, b) (a > b ? a : b)**

**int main()**

**{**

**int n, x, y, w;**

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

**while(scanf("%d%d%d", &x, &y, &w) == 3 && x && y && w)**

**{**

**map[x][y] = w;**

**}**

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

**{**

**for(int j = 1; j <= n; j++)**

**{**

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

**{**

**for(int l = 1; l <= n; l++)**

**{**

**int & tmp = dp[i][j][k][l];**

**int p = max(max(dp[i-1][j][k-1][l], dp[i-1][j][k][l-1]), max(dp[i][j-1][k-1][l], dp[i][j-1][k][l-1]));**

**tmp += (i == k && j == l) ? (p + map[i][j]) : (p + map[i][j] + map[k][l]);**

**}**

**}**

**}**

**}**

**printf("%d\n", dp[n][n][n][n]);**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define max(a,b) a>b?a:b**

**#define min(a,b) a<b?a:b**

**int main()**

**{**

**int map[11][11]={0},f[11][11]={0};**

**int i,j,k,N,t;**

**scanf("%d",&N);**

**while (scanf("%d%d%d",&i,&j,&k)&&(i||j||k)) map[i][j]=k;**

**for (i=2;i<=2\*N;i++)**

**for (t=min(i,N),j=t;j>0;j--)**

**for (k=t;k>0;k--)**

**{**

**f[j][k]=max(f[j][k],f[j-1][k-1]);**

**f[j][k]=max(f[j][k],f[j][k-1]);**

**f[j][k]=max(f[j][k],f[j-1][k]);**

**if (j==k) f[j][k]+=map[j][i-j];**

**else f[j][k]+=map[j][i-j]+map[k][i-k];**

**}**

**printf("%d",f[N][N]);**

**return 0;**

**}**

**JAVA**

**import java.io.BufferedReader;**

**import java.io.IOException;**

**import java.io.InputStreamReader;**

**public class Main{**

**static int x;**

**static int y;**

**static int n;**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**n = Integer.parseInt(br.readLine());**

**int[][] tag = new int[2 \* n + 1][2 \* n + 1];**

**int[][] arr = new int[n \* n][2];**

**out: for (int i = 1; ; i++) {**

**String[] str = br.readLine().split(" ");**

**for (int j = 0; j < 1; j++) {**

**x = arr[i][0] = Integer.parseInt(str[0]);**

**y = arr[i][1] = Integer.parseInt(str[1]);**

**tag[x][y] = Integer.parseInt(str[2]);**

**if (x == 0 && y == 0 && tag[x][y] == 0){**

**dp(tag);**

**break out;**

**}**

**}**

**}**

**}**

**public static void dp(int[][] tag) {**

**int[][] temp = new int[2 \* n][2 \* n];**

**int k;**

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

**for (int t = min(i, n), j = t; j > 0; j--) {**

**for (k = t; k > 0; k--) {**

**temp[j][k] = max(temp[j][k], temp[j - 1][k - 1]);**

**temp[j][k] = max(temp[j][k], temp[j - 1][k]);**

**temp[j][k] = max(temp[j][k], temp[j][k - 1]);**

**if (j == k)**

**temp[j][k] += tag[j][i - j];**

**else**

**temp[j][k] += tag[j][i - j] + tag[k][i - k];**

**}**

**}**

**}**

**System.out.println(temp[n][n]);**

**}**

**public static int max(int a, int b) {**

**return a > b ? a : b;**

**}**

**public static int min(int a, int b) {**

**return a > b ? b : a;**

**}**

**}**

ALGO-18**算法训练 单词接龙**

时间限制：1.0s   内存限制：256.0MB

关键字：搜索

问题描述

　　单词接龙是一个与我们经常玩的成语接龙相类似的游戏，现在我们已知一组单词，且给定一个开头的字母，要求出以这个字母开头的最长的“龙”（每个单词都最多在“龙”中出现两次），在两个单词相连时，其重合部分合为一部分，例如 beast和astonish，如果接成一条龙则变为beastonish，另外相邻的两部分不能存在包含关系，例如at 和 atide 间不能相连。

输入格式

　　输入的第一行为一个单独的整数n (n<=20)表示单词数，以下n 行每行有一个单词，输入的最后一行为一个单个字符，表示“龙”开头的字母。你可以假定以此字母开头的“龙”一定存在.

输出格式

　　只需输出以此字母开头的最长的“龙”的长度

样例输入

5  
at  
touch  
cheat  
choose  
tact  
a

样例输出

23

样例说明

　　连成的“龙”为atoucheatactactouchoose

**参考代码：**

**C++**

**#include <cstdio>**

**#include <cmath>**

**#include <cstring>**

**#include <iostream>**

**#include <algorithm>**

**using namespace std;**

**using namespace std ;**

**int map[25][25];**

**int n,cnt;**

**char c;**

**char st[25][25];**

**int visit[25];**

**int dfs(int s)**

**{**

**int temp,maxlen=0,t=0;**

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

**{**

**temp=0;**

**if(map[s][i]&&visit[i])**

**{**

**t=1;**

**visit[i]--;**

**temp=dfs(i);**

**visit[i]++;**

**temp=temp+strlen(st[s])-map[s][i];**

**}**

**if(temp>maxlen) maxlen=temp;**

**}**

**if(!t) return strlen(st[s]);**

**return maxlen;**

**}**

**int main()**

**{**

**while(scanf("%d",&n)!=EOF)**

**{**

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

**scanf("%s",st[i]);**

**getchar();**

**scanf("%c",&c);**

**memset(map,0,sizeof(map));**

**int len1,len2;**

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

**for(int j=0;j<n;j++)**

**{**

**len1=strlen(st[i]);**

**len2=strlen(st[j]);**

**for(int k=0;k<len1&&k<len2;k++)**

**{**

**if(strncmp(st[i]+len1-k-1,st[j],k+1)==0)**

**{**

**map[i][j]=k+1;**

**break;**

**}**

**}**

**}**

**for(int i=0;i<n;i++) visit[i]=2;**

**cnt=0;**

**int temp;**

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

**{**

**if(st[i][0]==c)**

**{**

**visit[i]--;**

**temp=dfs(i);**

**if(temp>cnt) cnt=temp;**

**visit[i]++;**

**}**

**}**

**printf("%d\n",cnt);**

**}**

**}**

**C**

**#include <stdio.h>**

**#include <string.h>**

**int Len[21],P[21],MAX,N;**

**char Word[21][100];**

**void DFS(int d,int s)**

**{**

**int i,j,k,l;**

**for (i=1;i<=N;i++)**

**if (P[i]<2)**

**for (j=0;j<Len[d];j++)**

**if (Word[i][0]==Word[d][j])**

**{**

**for (l=1,k=j+1;k<Len[d]&&Word[i][l]==Word[d][k];k++,l++);**

**if (k<Len[d]) continue;**

**P[i]++;**

**DFS(i,s+Len[i]-l);**

**P[i]--;**

**}**

**if (s>MAX) MAX=s;**

**}**

**int main()**

**{**

**int i;**

**scanf("%d",&N);**

**for (i=1;i<=N;Len[i]=strlen(Word[i]),i++) scanf("%s",Word[i]);**

**scanf("%s",Word[0]);**

**Len[0]=strlen(Word[0]);**

**DFS(0,Len[0]);**

**printf("%d",MAX);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**private static String[] a = new String[20];**

**private static int[] b = new int[20];**

**private static int max;**

**private static int n;**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**n = scanner.nextInt();**

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

**a[i] = scanner.next();**

**String string = scanner.next();**

**f(string, string.length());**

**System.out.println(max);**

**}**

**private static void f(String s, int length) {**

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

**if (a[i].indexOf(s) == 0 && b[i] < 2) {**

**int length1 = s.length();**

**int length2 = a[i].length();**

**b[i]++;**

**int p = 1;**

**length = length + length2 - length1;**

**while (p < length2) {**

**f(a[i].substring(length2 - p, length2), length);**

**p += 1;**

**}**

**length = length - length2 + length1;**

**b[i]--;**

**}**

**max = length > max ? length : max;**

**}**

**}**

ALGO-17**算法训练 乘积最大**

时间限制：1.0s   内存限制：256.0MB

关键字：动态规划

问题描述

　　今年是国际数学联盟确定的“2000——世界数学年”，又恰逢我国著名数学家华罗庚先生诞辰90周年。在华罗庚先生的家乡江苏金坛，组织了一场别开生面的数学智力竞赛的活动，你的一个好朋友XZ也有幸得以参加。活动中，主持人给所有参加活动的选手出了这样一道题目：  
  
　　设有一个长度为N的数字串，要求选手使用K个乘号将它分成K+1个部分，找出一种分法，使得这K+1个部分的乘积能够为最大。  
  
　　同时，为了帮助选手能够正确理解题意，主持人还举了如下的一个例子：  
  
　　有一个数字串：312， 当N=3，K=1时会有以下两种分法：  
  
　　3\*12=36  
　　31\*2=62  
  
　　这时，符合题目要求的结果是：31\*2=62  
  
　　现在，请你帮助你的好朋友XZ设计一个程序，求得正确的答案。

输入格式

　　程序的输入共有两行：  
　　第一行共有2个自然数N，K（6≤N≤40，1≤K≤6）  
　　第二行是一个长度为N的数字串。

输出格式

　　输出所求得的最大乘积（一个自然数）。

样例输入

4 2  
1231

样例输出

62

**参考代码：**

**C++**

**#include <iostream>**

**#include <string>**

**#include <cmath>**

**#include <cstring>**

**using namespace std;**

**int N,K;**

**string n;**

**int g[41] = { 0 };**

**long long int F[41][31] = { 0 };**

**inline long long int MAX(long long int a, long long int b)**

**{**

**return a>b?a:b;**

**}**

**long long int aget(int b, int c)**

**{**

**long long int temp = 0;**

**for(int i = b; i <= c; i++)**

**{**

**temp = (temp\*10 + g[i]);**

**}**

**return temp;**

**}**

**int main()**

**{**

**cin >> N >> K >> n;**

**K++;**

**char buf[N];**

**strcpy(buf,n.c\_str());**

**for(int i = 0; i< N; i++) g[i+1] = buf[i] - '0';**

**for(int i = 1; i <= N; i ++) F[i][1] = aget(1, i );**

**for(int j = 2; j <= K; j++)**

**for(int i = 1; i <= N; i++)**

**for(int k = 1; k < i; k++)**

**F[i][j] = MAX( F[i][j] , F[k][j-1]\*aget(k+1,i) );**

**cout << F[N][K];**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define maxN 41**

**#define maxK 7**

**#define InfiniteMin -999999999**

**int main()**

**{**

**int N,K;**

**int i,j,k,m;**

**int A[maxN][maxK]; /\*A[i][j]表示前i个数有j个乘号能达到的最大乘积\*/**

**int s[maxN];**

**char num[maxN];**

**int temp,max;**

**scanf("%d%d%s",&N,&K,num);**

**for(i=0;i<N;i++)**

**s[i+1]=num[i]-'0';**

**for(i=1;i<=N;i++)**

**{**

**temp=0;**

**for(j=1;j<=i;j++)**

**temp=temp\*10+s[j];**

**A[i][0]=temp;**

**}**

**for(j=1;j<=K;j++)**

**{**

**for(i=j+1;i<=N;i++)**

**{**

**max=InfiniteMin;**

**for(k=i;k-1>j-1;k--)**

**{**

**temp=0;**

**for(m=k;m<=i;m++)**

**temp=temp\*10+s[m];**

**temp\*=A[k-1][j-1];**

**max=max>temp?max:temp;**

**}**

**A[i][j]=max;**

**}**

**}**

**printf("%d",A[N][K]);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**private static int n, k;**

**private static char[] num;**

**private static int getOtherPart(int start, int end) {**

**int temp = 0;**

**for (; start < end; ++start) {**

**temp = temp \* 10 + num[start] - '0';**

**}**

**return temp;**

**}**

**private static int dp(int n, int k) {**

**if (k == 0) {**

**return getOtherPart(0, n);**

**}**

**int max = Integer.MIN\_VALUE;**

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

**int temp = dp(i, k - 1) \* getOtherPart(i, n);**

**max = Math.max(max, temp);**

**}**

**return max;**

**}**

**public static void main(String[] args) {**

**Scanner sc = new Scanner(System.in);**

**n = sc.nextInt();**

**k = sc.nextInt();**

**num = sc.next().toCharArray();**

**System.out.println(dp(n, k));**

**}**

**}**

ALGO-16**算法训练 进制转换**

时间限制：1.0s   内存限制：256.0MB

关键字：进制转换

问题描述

　　我们可以用这样的方式来表示一个十进制数： 将每个阿拉伯数字乘以一个以该数字所处位置的（值减１）为指数，以１０为底数的幂之和的形式。例如：１２３可表示为 １＊１０２＋２＊１０１＋３＊１００这样的形式。  
　　与之相似的，对二进制数来说，也可表示成每个二进制数码乘以一个以该数字所处位置的（值－１）为指数，以２为底数的幂之和的形式。一般说来，任何一个正整数Ｒ或一个负整数－Ｒ都可以被选来作为一个数制系统的基数。如果是以Ｒ或－Ｒ为基数，则需要用到的数码为 ０，１，．．．．Ｒ－１。例如，当Ｒ＝７时，所需用到的数码是０，１，２，３，４，５和６，这与其是Ｒ或－Ｒ无关。如果作为基数的数绝对值超过１０，则为了表示这些数码，通常使用英文字母来表示那些大于９的数码。例如对１６进制数来说，用Ａ表示１０，用Ｂ表示１１，用Ｃ表示１２，用Ｄ表示１３，用Ｅ表示１４，用Ｆ表示１５。  
　　在负进制数中是用－Ｒ 作为基数，例如－１５（十进制）相当于１１０００１（－２进制），并且它可以被表示为２的幂级数的和数：  
　　１１０００１＝１＊（－２）５＋１＊（－２）４＋０＊（－２）３＋０＊（－２）２＋  
　　０＊（－２）１ ＋１＊（－２）０  
　　设计一个程序，读入一个十进制数和一个负进制数的基数, 并将此十进制数转换为此负进制下的数： －Ｒ∈｛－２，－３，－４，．．．，－２０｝

输入格式

　　一行两个数，第一个是十进制数Ｎ（－32768＜＝Ｎ＜＝32767）， 第二个是负进制数的基数－Ｒ。

输出格式

　　输出所求负进制数及其基数，若此基数超过１０，则参照１６进制的方式处理。（格式参照样例）

样例输入

30000 -2

样例输出

30000=11011010101110000(base-2)

样例输入

-20000 -2

样例输出

-20000=1111011000100000(base-2)

样例输入

28800 -16

样例输出

28800=19180(base-16)

样例输入

-25000 -16

样例输出

-25000=7FB8(base-16)

**参考代码：**

**C++**

**#include <iostream>**

**#include <cstdio>**

**using namespace std;**

**int N,M,base;**

**inline int Div(int a,int b)**

**{**

**int n;**

**n = a / b;**

**if(n\*b <= a)**

**return n;**

**return n + 1;**

**}**

**void work()**

**{**

**int num[100];**

**int top = 0;**

**if(N == 0)**

**{**

**cout << 0;**

**}**

**int P;**

**while(N)**

**{**

**P = Div(N,base);**

**num[++top] = N - P\*base;**

**N = P;**

**}**

**for(;top >= 1;top--)**

**{**

**if(num[top]<10)**

**cout << num[top];**

**if(num[top] >= 10)**

**cout << (char)(num[top] - 10 + 'A');**

**}**

**cout << "(base" << base << ")" << endl;**

**}**

**int main()**

**{;**

**while(cin >> N)**

**{**

**cin >> base;**

**M = N;**

**cout << N << "=";**

**work();**

**}**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**const char nc[20]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F','G','H','I','J'};**

**char str[20];**

**int main()**

**{**

**int m, n, k, t, s;**

**int i=0;**

**scanf("%d%d",&m,&n);**

**s = m;**

**while(m != 0)**

**{**

**k = m % n;**

**t = m / n;**

**if(k < 0)**

**{**

**k -= n;**

**t++;**

**}**

**m = t;**

**str[i++]=nc[k];**

**}**

**printf("%d=",s);**

**for(i = i- 1; i >= 0; i--)**

**printf("%c",str[i]);**

**printf("(base%d)\n",n);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**int N = scanner.nextInt();**

**int R = scanner.nextInt();**

**char[] c = "0123456789ABCDEFG".toCharArray();**

**String s1 = N + "=";**

**String s = "";**

**while (N != 0) {**

**int t = N % R;**

**if (t < 0) {**

**t = t - R;**

**N = N / R + 1;**

**} else**

**N = N / R;**

**s = c[t] + s;**

**}**

**System.out.println(s1 + s + "(base" + R + ")");**

**}**

**}**

ALGO-15**算法训练 旅行家的预算**

时间限制：1.0s   内存限制：256.0MB

关键字：贪心

问题描述

　　一个旅行家想驾驶汽车以最少的费用从一个城市到另一个城市（假设出发时油箱是空的）。给定两个城市之间的距离D1、汽车油箱的容量C（以升为单位）、每升汽油能行驶的距离D2、出发点每升汽油价格P和沿途油站数N（N可以为零），油站i离出发点的距离Di、每升汽油价格Pi（i=1，2，……N）。计算结果四舍五入至小数点后两位。如果无法到达目的地，则输出“No Solution”。

输入格式

　　第一行为4个实数D1、C、D2、P与一个非负整数N；  
　　接下来N行，每行两个实数Di、Pi。

输出格式

　　如果可以到达目的地，输出一个实数（四舍五入至小数点后两位），表示最小费用；否则输出“No Solution”（不含引号）。

样例输入

275.6 11.9 27.4 2.8 2  
102.0 2.9  
220.0 2.2

样例输出

26.95

**参考代码：**

**C++**

**#include<cstdio>**

**using namespace std;**

**int main(){**

**int n;**

**double d1,c,d2,p0;**

**double p[1000],d[1000];**

**scanf("%lf%lf%lf%lf%d",&d1,&c,&d2,&p0,&n);**

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

**scanf("%lf%lf",&d[i],&p[i]);**

**}**

**d[0]=0;**

**p[0]=p0;**

**d[n+1]=d1;**

**p[n+1]=0;**

**double sum=0,ct=0;**

**bool flag=1;**

**for(int i=0,j;i<=n;i=j){**

**flag=0;**

**if(d[i+1]-d[i]>c\*d2)break;**

**flag=1;**

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

**if(d[j]-d[i]>c\*d2){**

**j--;**

**break;**

**}**

**if(p[j]<=p[i]) break;**

**}**

**if(p[j]<=p[i]){**

**sum+=((d[j]-d[i])/d2-ct)\*p[i];**

**ct=0;**

**}else{**

**sum+=(c-ct)\*p[i];**

**ct=c-(d[j]-d[i])/d2;**

**}**

**}**

**if(flag)printf("%.2lf\n",sum);**

**else printf("No Solution\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define MAX\_N 100**

**float D1,C,D2,P;**

**float A[MAX\_N],B[MAX\_N];**

**int N;**

**void solve()**

**{**

**int i;**

**float res = 0;**

**float pos,tank;**

**pos = B[0];**

**tank = C;**

**res += pos\*tank;**

**float we=0;**

**for(i=1; i<N; i++)**

**{**

**tank -= (A[i] - A[i-1]-we)/D2;**

**if(B[i] < pos)**

**{**

**we = 0;**

**res -= tank\*pos;**

**tank = C;**

**pos = B[i];**

**res += tank\*pos;**

**}**

**else**

**{**

**if(tank < (A[i+1]-A[i])/D2)**

**{**

**we=0;**

**we = tank\*D2;**

**tank = C;**

**pos = B[i];**

**res += tank\*pos;**

**}**

**if(tank < 0)**

**{**

**puts("No Solution\n");**

**return ;**

**}**

**}**

**}**

**tank -= (D1-A[i-1]-we)/D2;**

**if(tank < 0)**

**{**

**puts("No Solution\n");**

**return ;**

**}**

**res -= tank\*pos;**

**printf("%.2f\n",res);**

**}**

**int main()**

**{**

**int i;**

**scanf("%f%f%f%f%d",&D1,&C,&D2,&P,&N);**

**A[0] = 0;**

**B[0] = P;**

**N++;**

**A[N] = D1;**

**for(i=1; i<N; i++)**

**scanf("%f%f",&A[i],&B[i]);**

**if(D1 != 0)**

**solve();**

**else**

**puts("No Solution\n");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**double[][] p = new double[1024][1024];**

**Scanner sc = new Scanner(System.in);**

**int i, j, k = 0;**

**double d1 = sc.nextDouble();**

**double c = sc.nextDouble();**

**double d2 = sc.nextDouble();**

**p[0][1] = sc.nextDouble();**

**int n = sc.nextInt();**

**n++;**

**for (i = 1; i < n; i++) {**

**p[i][0] = sc.nextDouble();**

**p[i][1] = sc.nextDouble();**

**}**

**p[n++][0] = d1;**

**double f = c \* d2;**

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

**if (p[i + 1][0] - p[i][0] > f) {**

**System.out.println("No Solution");**

**return;**

**}**

**}**

**double min = 0, max, d;**

**for (i = 0; i < n - 1; i++) {**

**d = p[i + 1][0] - p[i][0];**

**while (d > 0) {**

**while (p[i + 1][0] - p[k][0] - d >= f)**

**k++;**

**for (j = k; j <= i; j++)**

**if (p[j][1] < p[k][1])**

**k = j;**

**max = f - (p[i + 1][0] - p[k][0] - d);**

**if (max > d)**

**max = d;**

**d -= max;**

**min += max / d2 \* p[k][1];**

**}**

**}**

**System.out.println(String.format("%.2f", min));**

**}**

**}**

ALGO-14**算法训练 回文数**

时间限制：1.0s   内存限制：256.0MB

关键字：模拟 高精度计算

问题描述

　　若一个数（首位不为零）从左向右读与从右向左读都一样，我们就将其称之为回文数。  
　　例如：给定一个10进制数56，将56加65（即把56从右向左读），得到121是一个回文数。  
  
　　又如：对于10进制数87：  
　　STEP1：87+78 = 165 STEP2：165+561 = 726  
　　STEP3：726+627 = 1353 STEP4：1353+3531 = 4884  
  
　　在这里的一步是指进行了一次N进制的加法，上例最少用了4步得到回文数4884。  
  
　　写一个程序，给定一个N（2<=N<=10或N=16）进制数M（其中16进制数字为0-9与A-F），求最少经过几步可以得到回文数。  
　　如果在30步以内（包含30步）不可能得到回文数，则输出“Impossible!”

输入格式

　　两行，N与M

输出格式

　　如果能在30步以内得到回文数，输出“STEP=xx”（不含引号），其中xx是步数；否则输出一行”Impossible!”（不含引号）

样例输入

9  
87

样例输出

STEP=6

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstring>**

**#include<iostream>**

**using namespace std;**

**struct bign{**

**int s[1000];**

**int n;**

**int len;**

**bign(){**

**memset(s,0,sizeof(s));**

**n=10;**

**len=1;**

**}**

**bign(int n,char \*m){**

**int l=strlen(m);**

**len=l;**

**memset(s,0,sizeof(s));**

**this->n=n;**

**for(int i=l-1,j=0;i>=0;i--){**

**if(m[i]>='0' && m[i]<='9')**

**s[j++]=m[i]-'0';**

**else**

**s[j++]=m[i]-'A'+10;**

**}**

**}**

**bign operator+(const bign&x)const{**

**bign ans;**

**int len=this->len>x.len?this->len:x.len+1;**

**int n=x.n;**

**for(int i=0;i<len;i++){**

**ans.s[i]+=x.s[i]+this->s[i];**

**ans.s[i+1]=ans.s[i]/n;**

**ans.s[i]%=n;**

**}**

**if(len>1 && ans.s[len-1]==0)len--;**

**ans.n=n;**

**ans.len=len;**

**return ans;**

**}**

**bign operator=(const bign&x){**

**this->len=x.len;**

**this->n=x.n;**

**for(int i=0;i<x.len;i++){**

**this->s[i]=x.s[i];**

**}**

**return \*this;**

**}**

**bign reversal()const{**

**bign ans;**

**ans.len=this->len;**

**ans.n=this->n;**

**for(int i=this->len-1,j=0;i>=0;i--){**

**ans.s[j++]=this->s[i];**

**}**

**return ans;**

**}**

**bool isPalindromes()const{**

**int i=this->len-1;**

**int j=0;**

**while(i>j){**

**if(this->s[i]!=this->s[j])break;**

**i--;**

**j++;**

**}**

**return i<=j;**

**}**

**};**

**ostream&operator<<(ostream&out,const bign&x){**

**for(int i=x.len-1;i>=0;i--){**

**if(x.s[i]<10)**

**out<<x.s[i];**

**else**

**out<<x.s[i]-10+'A';**

**}**

**return out;**

**}**

**int main(){**

**int n;**

**char m[1000];**

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

**getchar();**

**gets(m);**

**bign a(n,m);**

**int i=0;**

**while(!a.isPalindromes()){**

**if(i>30)break;**

**i++;**

**bign b=a.reversal();**

**a=a+b;**

**}**

**if(i<=30)printf("STEP=%d\n",i);**

**else printf("Impossible!\n");**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**#include<math.h>**

**#include <string.h>**

**#include <stdlib.h>**

**int p;**

**char s[1000];**

**int c[1000];**

**int reverse(int a[],int n) /\*判断a是否是回文\*/**

**{**

**int i,j;**

**i=0;**

**j=n-1;**

**while(i<j&&a[i]==a[j])**

**{**

**i++;**

**j--;**

**}**

**return i>=j;**

**}**

**int aplus(int a[],int n,int m) /\*计算a=a+a\*/**

**{**

**int \*b,i,j,kc;**

**b=(int \*)malloc(sizeof(int)\*1001);**

**for(i=0;i<n;i++)**

**b[n-1-i]=a[i];**

**kc=0;**

**for(i=0;i<n;i++)**

**{**

**a[i]=a[i]+b[i]+kc;**

**kc=a[i]/m;**

**a[i]=a[i]%m;**

**}**

**if(kc>0)**

**a[n++]=kc;**

**free(b);**

**return n;**

**}**

**int stod(char s[],int a[])**

**{**

**int i;**

**for(i=0;s[i]!='\0';i++)**

**if(s[i]>='A'&&s[i]<='F')**

**a[i]=10+s[i]-'A';**

**else**

**a[i]=s[i]-'0';**

**return i;**

**}**

**int main()**

**{**

**int n,i;**

**scanf("%d",&p);**

**scanf("%s",s);**

**n=stod(s,c);**

**for(i=0;i<30;i++)**

**{**

**n=aplus(c,n,p);**

**if(reverse(c,n)==1)**

**{**

**printf("STEP=%d\n",i+1);**

**break;**

**}**

**}**

**if(i>=30)**

**printf("Impossible!\n");**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**private static int sum;**

**private static int n;**

**public static void main(String[] args) {**

**Scanner scanner = new Scanner(System.in);**

**n = scanner.nextInt();**

**String M = scanner.next();**

**f(M, 0);**

**if (sum <= 30)**

**System.out.println("STEP="+sum);**

**else**

**System.out.println("Impossible!");**

**}**

**private static void f(String m, int i) {**

**StringBuffer stringBuffer1 = new StringBuffer(m);**

**StringBuffer stringBuffer2 = new StringBuffer(m).reverse();**

**if (i > 30)**

**return;**

**long l1 = Long.parseLong(stringBuffer1.toString(), n);**

**long l2 = Long.parseLong(stringBuffer2.toString(), n);**

**if (l1 == l2)**

**return;**

**sum++;**

**f(change(l1 + l2), i + 1);**

**}**

**public static String change(long l) {**

**if (n == 16)**

**return Long.toHexString(l);**

**StringBuffer buf = new StringBuffer();**

**for (; l >= n;) {**

**buf.insert(0, l % n);**

**l = l / n;**

**}**

**buf.insert(0, l % n);**

**return buf.toString();**

**}**

**}**

ALGO-13**算法训练 拦截导弹**

时间限制：1.0s   内存限制：256.0MB

关键字：贪心 动态规划

问题描述

　　某国为了防御敌国的导弹袭击，发展出一种导弹拦截系统。但是这种导弹拦截系统有一个缺陷：虽然它的第一发炮弹能够到达任意的高度，但是以后每一发炮弹都不能高于前一发的高度。某天，雷达捕捉到敌国的导弹来袭。由于该系统还在试用阶段，所以只有一套系统，因此有可能不能拦截所有的导弹。  
  
　　输入导弹依次飞来的高度（雷达给出的高度数据是不大于30000的正整数），计算这套系统最多能拦截多少导弹，如果要拦截所有导弹最少要配备多少套这种导弹拦截系统。

输入格式

　　一行，为导弹依次飞来的高度

输出格式

　　两行，分别是最多能拦截的导弹数与要拦截所有导弹最少要配备的系统数

样例输入

389 207 155 300 299 170 158 65

样例输出

6  
2

**参考代码：**

**C++**

**#include <cstdio>**

**#include <cstdlib>**

**#include <cstring>**

**using namespace std;**

**int a[10001],high[10001];**

**int n = 1, top, maw;**

**void work(int p)**

**{**

**int l = 1, r = n, mid;**

**while(l != r)**

**{**

**mid = (l + r) / 2;**

**if(p < high[mid]) l = mid + 1;**

**else r = mid;**

**}**

**if(p > high[l]) high[l] = p;**

**if(l > maw) maw = l;**

**}**

**void solve(int p)**

**{**

**int l = 1, r = n, mid;**

**while(l != r)**

**{**

**mid = (l + r) / 2;**

**if(p > high[mid]) l = mid + 1;**

**else r = mid;**

**}**

**if(p < high[l]) high[l] = p;**

**if(l > top) top = l;**

**}**

**int main()**

**{**

**//freopen("missile.txt","r",stdin);**

**//freopen("missiles.txt","w",stdout);**

**while(scanf("%d", &a[n]) != EOF) ++n;**

**--n;**

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

**work(a[i]);**

**memset(high, 1, sizeof(high));**

**for(int i = 1; i <= n; ++i) solve(a[i]);**

**printf("%d\n%d\n", maw, top);**

**// system("pause");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define MAX\_N 10000**

**int dp[MAX\_N]={0};**

**int i=0;**

**int dao[MAX\_N];**

**int max(int a,int b)**

**{**

**return a>b?a:b;**

**}**

**void solve1()**

**{**

**int k,j;**

**int res = 0;**

**int n = i;**

**for(j=0; j<n; j++)**

**{**

**dp[j] = 1;**

**for(k=0; k<j; k++)**

**if(dao[k] > dao[j])**

**dp[j] = max(dp[j],dp[k]+1);**

**res = max(res , dp[j]);**

**}**

**printf("%d\n",res);**

**}**

**void solve2()**

**{**

**int k,j;**

**int res = 0;**

**int n = i;**

**for(j=0; j<n; j++)**

**{**

**dp[j] = 1;**

**for(k=0; k<j; k++)**

**if(dao[k] < dao[j])**

**dp[j] = max(dp[j],dp[k]+1);**

**res = max(res , dp[j]);**

**}**

**printf("%d\n",res);**

**}**

**int main()**

**{**

**char q;**

**int s=0;**

**while(q=getchar())**

**{**

**if(q>='0' && q<='9')**

**s = s\*10+q-'0';**

**else if(q == ' ')**

**{**

**dao[i] = s;**

**i++;**

**s = 0;**

**}**

**else**

**break;**

**}**

**dao[i] = s;**

**i++;**

**solve1();**

**solve2();**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.\*;**

**public class Main {**

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

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

**String s=bf.readLine();**

**String[] ss=s.split(" ");**

**int[] numa=new int[ss.length];**

**int[] numb=new int[ss.length];**

**int[] numc=new int[ss.length];**

**for(int i=0;i<ss.length;i++){**

**numa[i]=Integer.parseInt(ss[i]);**

**numb[i]=1;**

**numc[i]=1;**

**}**

**int a1=Integer.MIN\_VALUE;**

**int a2=Integer.MIN\_VALUE;**

**for(int i=0;i<numa.length;i++){**

**for(int j=0;j<i;j++){**

**if(numa[i]<numa[j]&&numb[i]<numb[j]+1){**

**numb[i]=numb[j]+1;**

**}**

**a1=Math.max(a1, numb[i]);**

**}**

**}**

**for(int i=0;i<numa.length;i++){**

**for(int j=0;j<i;j++){**

**if(numa[i]>numa[j]&&numc[i]<numc[j]+1){**

**numc[i]=numc[j]+1;**

**}**

**}**

**a2=Math.max(a2, numc[i]);**

**}**

**System.out.println(a1);**

**System.out.println(a2);**

**}**

**}**

ALGO-12**算法训练 幂方分解**

时间限制：1.0s   内存限制：256.0MB

关键字：递归

问题描述

　　任何一个正整数都可以用2的幂次方表示。例如：  
　　137=27+23+20  
　　同时约定方次用括号来表示，即ab 可表示为a（b）。  
　　由此可知，137可表示为：  
　　2（7）+2（3）+2（0）  
　　进一步：7= 22+2+20（21用2表示）  
　　3=2+20  
　　所以最后137可表示为：  
　　2（2（2）+2+2（0））+2（2+2（0））+2（0）  
　　又如：  
　　1315=210 +28 +25 +2+1  
　　所以1315最后可表示为：  
　　2（2（2+2（0））+2）+2（2（2+2（0）））+2（2（2）+2（0））+2+2（0）

输入格式

　　输入包含一个正整数N（N<=20000），为要求分解的整数。

输出格式

　　程序输出包含一行字符串，为符合约定的n的0，2表示（在表示中不能有空格）

**参考代码：**

**C++**

**#include<cstdio>**

**#include<stack>**

**using namespace std;**

**void f(int n){**

**int i=0;**

**stack<int>s;**

**if(n<=2)**

**printf("%d",n);**

**else while(n>0){**

**if(n&1){**

**s.push(i);**

**}**

**n>>=1;**

**i++;**

**}**

**bool p=0;**

**while(!s.empty()){**

**if(p)printf("+");**

**p=1;**

**i=s.top();**

**s.pop();**

**if(i!=1){**

**printf("2(");**

**f(i);**

**printf(")");**

**}else printf("2");**

**}**

**}**

**int main(){**

**int n;**

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

**f(n);**

**printf("\n");**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**void f(int a)**

**{**

**int i=0,j,b[32],w,k;**

**if(a==0)printf("0");**

**else if(a==2)printf("2");**

**else if(a==1)printf("2(0)");**

**else**

**{**

**while(a){b[i]=a%2;a=a/2;i++;}w=i;**

**k=0;j=0;for(i=w-1;i>=0;i--)if(b[i])k++;**

**for(i=w-1;i>=0;i--)**

**if(b[i])**

**{j++;**

**if(i==1)printf("2");**

**else {printf("2(");f(i);printf(")");}**

**if(j!=k)printf("+");}**

**}**

**}**

**int main()**

**{**

**int a;scanf("%d",&a);**

**f(a);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main {**

**public static void main(String[] args) {**

**f(new Scanner(System.in).nextInt());**

**}**

**private static void f(int n) {**

**int l = log(n);**

**if (l == 1) {**

**System.out.print("2");**

**} else if (l == 0) {**

**System.out.print("2(0)");**

**} else {**

**System.out.print("2(");**

**f(l);**

**System.out.print(")");**

**}**

**int s = (int) Math.pow(2, l);**

**int x = n - s;**

**if (x > 0) {**

**System.out.print("+");**

**f(x);**

**}else {**

**return;**

**}**

**}**

**private static int log(int n) {**

**if (n == 1)**

**return 0;**

**else**

**return 1 + log(n >> 1);**

**}**

**}**

ALGO-11**算法训练 瓷砖铺放**

时间限制：1.0s   内存限制：512.0MB

关键字：递归

问题描述

　　有一长度为N(1<=Ｎ<=10)的地板，给定两种不同瓷砖：一种长度为1，另一种长度为2，数目不限。要将这个长度为N的地板铺满，一共有多少种不同的铺法？  
　　例如，长度为4的地面一共有如下5种铺法：  
　　4=1+1+1+1  
　　4=2+1+1  
　　4=1+2+1  
　　4=1+1+2  
　　4=2+2  
　　编程用递归的方法求解上述问题。

输入格式

　　只有一个数N，代表地板的长度

输出格式

　　输出一个数，代表所有不同的瓷砖铺放方法的总数

样例输入

4

样例输出

5

**参考代码：**

**C++**

**#include<cstdio>**

**#include<cstdlib>**

**int a[20];**

**int main()**

**{**

**int n;**

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

**a[0]=1; a[1]=1;**

**for(int i=2;i<=n;i++) a[i]=a[i-1]+a[i-2];**

**printf("%d\n",a[n]);**

**//system("pause");**

**}**

**C**

**#include <stdio.h>**

**int j(int a)**

**{**

**int i=1,s=1;**

**for(;i<=a;i++)s\*=i;**

**return s;**

**}**

**int main()**

**{**

**int a;scanf("%d",&a);**

**int s=0,b=0,i;**

**while(a>=0)**

**{**

**s+=j(a+b)/j(a)/j(b);**

**a-=2;b++;**

**}**

**printf("%d",s);**

**return 0;**

**}**

**JAVA**

**import java.util.Scanner;**

**public class Main**

**{**

**public static void main(String []args)**

**{**

**Scanner input = new Scanner(System.in);**

**int n = input.nextInt();**

**if(1<=n && n<=10)**

**{**

**if(n==1)**

**{**

**System.out.println("1");**

**}**

**else**

**{**

**int []tag = new int[n];**

**tag[0] = 1;**

**tag[1] = 2;**

**if(n==1)**

**{**

**System.out.println(tag[0]);**

**}**

**for(int a=2;a<n;a++)**

**{**

**tag[a] = tag[a-1] + tag[a-2];**

**}**

**System.out.println(tag[n-1]);**

**}**

**}**

**}**

**}**

ALGO-10**算法训练 集合运算**

时间限制：1.0s   内存限制：512.0MB

关键字：数组 排序

问题描述

　　给出两个整数集合A、B，求出他们的交集、并集以及B在A中的余集。

输入格式

　　第一行为一个整数n，表示集合A中的元素个数。  
　　第二行有n个互不相同的用空格隔开的整数，表示集合A中的元素。  
　　第三行为一个整数m，表示集合B中的元素个数。  
　　第四行有m个互不相同的用空格隔开的整数，表示集合B中的元素。  
　　集合中的所有元素均为int范围内的整数，n、m<=1000。

输出格式

　　第一行按从小到大的顺序输出A、B交集中的所有元素。  
　　第二行按从小到大的顺序输出A、B并集中的所有元素。  
　　第三行按从小到大的顺序输出B在A中的余集中的所有元素。

样例输入

5  
1 2 3 4 5  
5  
2 4 6 8 10

样例输出

2 4  
1 2 3 4 5 6 8 10  
1 3 5

样例输入

4  
1 2 3 4  
3  
5 6 7

样例输出

1 2 3 4 5 6 7  
1 2 3 4

**参考代码：**

**C++**

**#include<cstdio>**

**#include<algorithm>**

**using namespace std;**

**int ta[2005];**

**int main()**

**{**

**int n,m,a[2005],b[2005];**

**int i,j,k;**

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

**for(i=0;i<n;i++)**

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

**scanf("%d",&m);**

**for(i=0;i<m;i++)**

**scanf("%d",&b[i]);**

**sort(a,a+n);**

**sort(b,b+m);**

**k=0;**

**for(i=0;i<n;i++)**

**{**

**for(j=0;j<m;j++)**

**{**

**if(a[i]<=b[j])**

**break;**

**}**

**if(j!=m)**

**{**

**if(a[i]==b[j])**

**{**

**ta[i]=-1;**

**k=1;**

**printf("%d ",a[i]);**

**}**

**}**

**}**

**if(k)**

**printf("\n");**

**i=0;j=0;**

**while(i<n&&j<m)**

**{**

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

**{**

**if(ta[i]!=-1)**

**printf("%d ",a[i]);**

**i++;**

**}**

**else**

**{**

**printf("%d ",b[j]);**

**j++;**

**}**

**}**

**for(i;i<n;i++)**

**{**

**if(ta[i]!=-1)**

**printf("%d ",a[i]);**

**}**

**for(j;j<m;j++)**

**{**

**printf("%d ",b[j]);**

**}**

**printf("\n");**

**k=0;**

**for(i=0;i<n;i++)**

**{**

**if(ta[i]!=-1)**

**{**

**k=1;**

**printf("%d ",a[i]);**

**}**

**}**

**if(k)**

**printf("\n");**

**return 0;**

**}**

**C**

**#include <stdio.h>**

**#define MAX\_N 1001**

**int n,m;**

**int Partition(int x[],int low,int high)**

**{**

**int pivotkey;**

**pivotkey = x[low];**

**while(low < high)**

**{**

**while(low<high && x[high]>=pivotkey)**

**high--;**

**x[low] = x[high];**

**while(low<high && x[low]<=pivotkey)**

**low++;**

**x[high] = x[low];**

**}**

**x[low] = pivotkey;**

**return low;**

**}**

**void QSort(int x[],int low,int high)**

**{**

**int pivotloc;**

**if(low < high)**

**{**

**pivotloc = Partition(x,low,high);**

**QSort(x,low,pivotloc-1);**

**QSort(x,pivotloc+1,high);**

**}**

**}**

**void jiao(int x[],int y[],int xx,int yy)**

**{**

**int i,j;**

**i = 1;**

**j = 1;**

**while(i<=xx && j<=yy)**

**{**

**if(x[i] == y[j])**

**{**

**printf("%d ",x[i]);**

**i++;**

**j++;**

**}**

**else if(x[i] < y[j])**

**i++;**

**else**

**j++;**

**}**

**printf("\n");**

**}**

**int bing(int x[],int y[],int z[],int xx,int yy)**

**{**

**int i,j,k;**

**i = 1;**

**j = 1;**

**k = 1;**

**while(i<=xx && j<=yy)**

**{**

**if(x[i] < y[j])**

**{**

**z[k] = x[i];**

**i++;**

**}**

**else if(x[i] > y[j])**

**{**

**z[k] = y[j];**

**j++;**

**}**

**else**

**{**

**z[k] = x[i];**

**i++;**

**j++;**

**}**

**k++;**

**}**

**if(i > xx)**

**while(j <= yy)**

**{**

**z[k] = y[j];**

**k++;**

**j++;**

**}**

**else**

**while(i <= xx)**

**{**

**z[k] = x[i];**

**k++;**

**i++;**

**}**

**return k;**

**}**

**void yu(int x[],int y[],int xx,int yy)**

**{**

**int i,j;**

**i = 1;**

**j = 1;**

**while(i<=xx && j<=yy)**

**{**

**if(x[i] == y[j])**

**{**

**i++;**

**j++;**

**}**

**else if(x[i] < y[j])**

**{**

**printf("%d ",x[i]);**

**i++;**

**}**

**else**

**j++;**

**}**

**if(j > yy)**

**while(i <= xx)**

**{**

**printf("%d ",x[i]);**

**i++;**

**}**

**printf("\n");**

**}**

**int main()**

**{**

**int i,l;**

**int a[MAX\_N],b[MAX\_N];**

**int c[2014];**

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

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

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

**QSort(a,1,n);**

**scanf("%d",&m);**

**for(i=1; i<=m; i++)**

**scanf("%d",&b[i]);**

**QSort(b,1,m);**

**jiao(a,b,n,m);**

**l = bing(a,b,c,n,m);**

**for(i=1; i<l; i++)**

**printf("%d ",c[i]);**

**printf("\n");**

**yu(a,b,n,m);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**import java.util.Arrays;**

**import java.util.HashSet;**

**import java.util.Iterator;**

**import java.util.Set;**

**public class Main {**

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

**BufferedReader br = new BufferedReader(new InputStreamReader(System.in));**

**int n = Integer.parseInt(br.readLine());**

**int[] arr = new int[n];**

**String st[] = br.readLine().split(" ");**

**for (int a = 0; a < arr.length; a++) {**

**arr[a] = Integer.parseInt(st[a]);**

**}**

**Arrays.sort(arr);**

**int m = Integer.parseInt(br.readLine());**

**int[] tag = new int[m];**

**String str[] = br.readLine().split(" ");**

**for (int a = 0; a < tag.length; a++) {**

**tag[a] = Integer.parseInt(str[a]);**

**}**

**Arrays.sort(tag);**

**func(arr, tag);**

**}**

**public static void func(int[] arr, int[] tag) {**

**int x;**

**for (int a = 0; a < arr.length; a++) {**

**x = Arrays.binarySearch(tag, arr[a]);**

**if (x >= 0) {**

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

**}**

**}**

**System.out.println();**

**Set<Integer> set = new HashSet<Integer>();**

**for (int a = 0; a < arr.length; a++) {**

**set.add(arr[a]);**

**}**

**for (int a = 0; a < tag.length; a++) {**

**set.add(tag[a]);**

**}**

**int[] sor = new int[set.size()];**

**Iterator<Integer> it = set.iterator();**

**while (it.hasNext()) {**

**for (int a = 0; a < sor.length; a++) {**

**sor[a] = it.next();**

**}**

**}**

**Arrays.sort(sor);**

**for (int a = 0; a < sor.length; a++) {**

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

**}**

**System.out.println();**

**int y;**

**for (int a = 0; a < arr.length; a++) {**

**y = Arrays.binarySearch(tag, arr[a]);**

**if (y < 0) {**

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

**}**

**}**

**System.out.println();**

**}**

**}**

ALGO-9**算法训练 摆动序列**

时间限制：1.0s   内存限制：512.0MB

关键字：动态规划

问题描述

　　如果一个序列满足下面的性质，我们就将它称为摆动序列：  
　　1. 序列中的所有数都是不大于k的正整数；  
　　2. 序列中至少有两个数。  
　　3. 序列中的数两两不相等；  
　　4. 如果第i – 1个数比第i – 2个数大，则第i个数比第i – 2个数小；如果第i – 1个数比第i – 2个数小，则第i个数比第i – 2个数大。  
　　比如，当k = 3时，有下面几个这样的序列：  
　　1 2  
　　1 3  
　　2 1  
　　2 1 3  
　　2 3  
　　2 3 1  
　　3 1  
　　3 2  
　　一共有8种，给定k，请求出满足上面要求的序列的个数。

输入格式

　　输入包含了一个整数k。（k<=20）

输出格式

　　输出一个整数，表示满足要求的序列个数。

样例输入

3

样例输出

8

**参考代码：**

**C++**

**#include<stdio.h>**

**int main()**

**{**

**int n,i,a[30],sum=0;**

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

**a[2]=2;**

**sum+=a[2];**

**for(i=3;i<=n;i++)**

**{**

**a[i]=sum+i\*(i-1);**

**sum+=a[i];**

**}**

**printf("%d",a[n]);**

**return 0;**

**}**

**C**

**#include<stdio.h>**

**int f[21][21][21]; //f[i][j][k] i表示数的长度，j表示倒数第2位的取值，k表示最后一位的取值**

**int main()**

**{**

**int n,i,j,k,p;**

**int sum=0;**

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

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

**for(j=1;j<=n;j++)**

**if(i!=j)**

**f[2][i][j]=1;**

**for(i=3;i<=n;i++)//从长度为3开始**

**{**

**for(j=1;j<=n;j++)**

**{**

**for(k=1;k<=n;k++)**

**{**

**for(p=1;p<=n;p++)**

**{**

**if(j>p&&k<p||j<p&&k>p)**

**f[i][j][k]+=f[i-1][p][j];**

**}**

**}**

**}**

**}**

**for(i=2;i<=n;i++)**

**for(j=1;j<=n;j++)**

**for(k=1;k<=n;k++)**

**sum+=f[i][j][k];**

**printf("%d",sum);**

**return 0;**

**}**

**JAVA**

**import java.io.\*;**

**public class Main{**

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

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

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

**System.out.println((int)(Math.pow(2,n)-n-1)\*2);**

**}**

**}**

ADV-112**算法提高 c++\_ch02\_01**

时间限制：1.0s   内存限制：512.0MB

编写一个程序，利用强制类型转换打印元音字母大小写10种形式的ASCII码。

输出的顺序为：大写的字母A，E，I，O，U的ASCII码，小写的字母a，e，i，o，u的ASCII码。所有的ASCII码都用十进制表示.输出10行,每行一个ASCII码，最后输出一个空行。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

cout<<65<<endl;

cout<<69<<endl;

cout<<73<<endl;

cout<<79<<endl;

cout<<85<<endl;

cout<<97<<endl;

cout<<101<<endl;

cout<<105<<endl;

cout<<111<<endl;

cout<<117<<endl;

return 0;

}

**C**

#include<stdio.h>

int main()

{

char a,e,i,o,u;

a='a';

e='e';

i='i';

o='o';

u='u';

printf("%d\n",a-32);

printf("%d\n",e-32);

printf("%d\n",i-32);

printf("%d\n",o-32);

printf("%d\n",u-32);

printf("%d\n",a);

printf("%d\n",e);

printf("%d\n",i);

printf("%d\n",o);

printf("%d\n",u);

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

System.out.println((int)'A');

System.out.println((int)'E');

System.out.println((int)'I');

System.out.println((int)'O');

System.out.println((int)'U');

System.out.println((int)'a');

System.out.println((int)'e');

System.out.println((int)'i');

System.out.println((int)'o');

System.out.println((int)'u');

System.out.println();

}

}

ADV-103**算法提高 逆序排列**

时间限制：1.0s   内存限制：512.0MB

关键字 循环语句 数组操作

问题描述

　　编写一个程序，读入一组整数（不超过20个），并把它们保存在一个整型数组中。当用户输入0时，表示输入结束。然后程序将把这个数组中的值按逆序重新存放，并打印出来。例如：假设用户输入了一组数据：7 19 -5 6 2 0，那么程序将会把前五个有效数据保存在一个数组中，即7 19 -5 6 2，然后把这个数组中的值按逆序重新存放，即变成了2 6 -5 19 7，然后把它们打印出来。

　　输入格式：输入只有一行，由若干个整数组成，中间用空格隔开，最末尾的整数为0。

　　输出格式：输出也只有一行，即逆序排列后的整数，中间用空格隔开，末尾没有空格。

　　输入输出样例

样例输入

7 19 -5 6 2 0

样例输出

2 6 -5 19 7

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int a[25];

int main()

{

int i=0,j,temp;

while(cin>>temp)

{

if(temp==0)break;

a[i]=temp;

i++;

}

if(i==0)cout<<endl;

else

{

for(j=i-1;j>0;j--)

cout<<a[j]<<" ";

cout<<a[0]<<endl;

}

return 0;

}

**C**

#include <stdio.h>

#include<string.h>

int main ()

{

int a[20],i,j,t;

for( i=0;i<20;i++)

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

if(a[i]==0)

{

break;

}

}

for(j=0;j<i/2;j++)

{

t=a[i-j-1];

a[i-j-1]=a[j];

a[j]=t;

}

for(j=0;j<i;j++)

{ printf("%d ",a[j]);

if(j==i-1)

printf("\b");

}

printf("\n");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int a[] = new int[21];

int n = 0;

for(int i=0;i<21;i++){

a[i] = scanner.nextInt();

if(a[i]==0){

break;

}

n++;

}

for(int i=n-1;i>=0;i--){

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

}

}

}

ADV-100**算法提高 第二大整数**

时间限制：1.0s   内存限制：512.0MB

关键字 逻辑判断 循环语句

问题描述

　　编写一个程序，读入一组整数（不超过20个），当用户输入0时，表示输入结束。然后程序将从这组整数中，把第二大的那个整数找出来，并把它打印出来。说明：（1）0表示输入结束，它本身并不计入这组整数中。（2）在这组整数中，既有正数，也可能有负数。（3）这组整数的个数不少于2个。  
　　输入格式：输入只有一行，包括若干个整数，中间用空格隔开，最后一个整数为0。  
　　输出格式：输出第二大的那个整数。  
　　输入输出样例

样例输入

5 8 -12 7 0

样例输出

7

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

using namespace std;

int main()

{

int i,fir,sec,n,map[21],m;

cin>>m;

n=0;

while(m)

{

map[n++]=m;

cin>>m;

}

if(map[0]>map[1])

{

fir=map[0];

sec=map[1];

}

else

{

fir=map[1];

sec=map[0];

}

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

{

if(map[i]>fir)

{

sec=fir;

fir=map[i];

}

if(map[i]<fir&&map[i]>sec)

sec=map[i];

}

cout<<sec<<endl;

return 0;

}

C

#include <stdio.h>

int main()

{

int t,mone,mtwo;

scanf("%d",&t);

mone=mtwo=-10000000;

while(t!=0)

{

if(t>mone)

{mtwo=mone,mone=t;}

else if(t>mtwo) mtwo=t;

scanf("%d",&t);

}

printf("%d",mtwo);

return 0;

}

**JAVA**

import java.util.\*;

import static java.lang.Math.\*;

/\*\*

\* Created by uulake on 11/24/14.

\*/

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

List<Integer> l = new ArrayList<>();

int t = 0;

while ((t = in.nextInt()) != 0){

l.add(t);

}

Collections.sort(l);

System.out.println(l.get(l.size()-2));

in.close();

}

}

ADV-98**算法提高 约数个数**

时间限制：1.0s   内存限制：512.0MB

输入一个正整数N (1

样例输入

12

样例输出

6

样例说明

12的约数包括：1,2,3,4,6,12。共6个

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main () {

int n;

while (cin >> n) {

int c = 0;

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

if (n % i == 0)

c++;

}

cout << c << endl;

}

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n,i,num=0;

scanf("%d",&n);

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

if(n%i==0) num++;

printf("%d",num);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int sum = 0;

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

{

if(n%i==0)

sum++;

}

System.out.println(sum);

}

}

ADV-97**算法提高 十进制数转八进制数**

时间限制：1.0s   内存限制：512.0MB

编写函数，其功能为把一个十进制数转换为其对应的八进制数。程序读入一个十进制数，调用该函数实现数制转换后，输出对应的八进制数。

样例输入

9274

样例输出

22072

样例输入

18

样例输出

22

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,b[50]={0},i=0;

scanf("%d",&a);

while(a)

{

b[i]=a%8;

a=a/8;

i++;

}

while(b[i]==0)

i--;

while(i>=0)

{

printf("%d",b[i]);

i--;

}

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n;

scanf("%d",&n);

printf("%o",n);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

sc.close();

System.out.println(Integer.toOctalString(n));

}

}

ADV-94**算法提高 复数归一化**

时间限制：1.0s   内存限制：512.0MB

编写函数Normalize，将复数归一化，即若复数为a+bi，归一化结果为a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b) 。使用结构体指针类型作为函数参数可能是必要的。其中实部和虚部由键盘输入，输出为归一化结果，如果归一化结果的实部或虚部为小数的要求保留一位小数。  
　　样例输入:（格式说明：3 4 分别为以空格隔开的实数的实部和虚部）  
　　3 4

样例输出

0.6+0.8i

样例输入

2 5

样例输出

0.4+0.9i

**参考代码：**

**C++**

#include <stdio.h>

#include <math.h>

typedef struct complex{

float real, imag;

}COMPLEX, \*COMPLEXP;

int main(void){

float e;

COMPLEX com;

scanf("%f %f",&(com.real),&(com.imag));

e = sqrt(com.imag \* com.imag + com.real \* com.real);

com.imag = com.imag/e;

com.real = com.real/e;

printf("%0.1f+%0.1fi",com.real,com.imag);

return 0;

}

**C**

#include <stdio.h>

#include <math.h>

//a/sqrt(a\*a+b\*b) + i\*b/sqrt(a\*a+b\*b)

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%.1f+%.1fi",a/sqrt(a\*a+b\*b),b/sqrt(a\*a+b\*b));

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s = new Scanner(System.in);

long a=s.nextInt();

long b=s.nextInt();

double r = a/Math.sqrt(a\*a+b\*b);

double i = b/Math.sqrt(a\*a+b\*b);

System.out.printf("%.1f+%.1fi\n", r, i);

}

}

ADV-15**算法提高 最大乘积**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　对于n个数，从中取出m个数，如何取使得这m个数的乘积最大呢？

输入格式

　　第一行一个数表示数据组数  
　　每组输入数据共2行：  
　　第1行给出总共的数字的个数n和要取的数的个数m，1<=n<=m<=15，  
　　第2行依次给出这n个数，其中每个数字的范围满足:a[i]的绝对值小于等于4。

输出格式

　　每组数据输出1行，为最大的乘积。

样例输入

15 51 2 3 4 5

样例输出

120

**参考代码：**

**C++**

#include "iostream"

#include "stdio.h"

#include "string"

#include "math.h"

#include "ctype.h"

#include "vector"

#include "stdlib.h"

#include "string.h"

#include "time.h"

#include "set"

#include "algorithm"

using namespace std;

int ans=-(1<<29);

int n,m;

int a[1000];

void dfs(int \*a,int cur,int sum,int k)

{

if(k>m)

return ;

if(cur==n)

{

if(k==m)

ans=std::max(ans,sum);

return ;

}

dfs(a,cur+1,sum\*a[cur],k+1);

dfs(a,cur+1,sum,k);

}

int main()

{

int T;

cin>>T;

while(T--)

{

cin>>n>>m;

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

{

cin>>a[i];

}

ans=-(1<<29);

dfs(a,0,1,0);

cout<<ans<<endl;

}

return 0;

}

**C**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int max;

void fun(int \*val,int index,int size,int m,int cnt,int res);

int main()

{

int n,m,x,i;

int val[15];

scanf("%d",&x);

while(x--)

{

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

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

{

scanf("%d",&val[i]);

}

max = -10000000;

fun(val,0,n,m,0,1);

printf("%d\n",max);

}

return 0;

}

void fun(int \*val,int index,int size,int m,int cnt,int res)

{

if(m==cnt)

{

if(res > max)

{

max = res;

}

return ;

}

if(index>=size)

{

return ;

}

fun(val,index+1,size,m,cnt+1,res\*val[index]);

fun(val,index+1,size,m,cnt,res);

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

int s = 0x80000000;

public void input() {

Scanner in = new Scanner(System.in);

int t = in.nextInt();

while (t > 0) {

s = 0x80000000;

t--;

int n = in.nextInt();

int m = in.nextInt();

int[] num = new int[n];

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

num[i] = in.nextInt();

}

count(num, 0, m, n,1);

int min = num[0];

System.out.println(s);

}

}

private void count(int[] num,int i,int m,int n,int sum){

if(m==0&&sum>s)s=sum;

if(i<n&&m>=0){

count(num,i+1,m-1,n,sum\*num[i]);

count(num,i+1,m,n,sum);

}

}

public static void main(String[] args) {

new Main().input();

}

}

ADV-5**算法提高 最小方差生成树**

时间限制：1.0s   内存限制：256.0MB

关键字 最小生成树

问题描述

给定带权无向图，求出一颗方差最小的生成树。

输入格式

输入多组测试数据。第一行为N,M，依次是点数和边数。接下来M行，每行三个整数U,V,W，代表连接U,V的边，和权值W。保证图连通。n=m=0标志着测试文件的结束。

输出格式

对于每组数据，输出最小方差，四舍五入到0.01。输出格式按照样例。

样例输入

4 5  
1 2 1  
2 3 2  
3 4 2  
4 1 1  
2 4 3  
4 6  
1 2 1  
2 3 2  
3 4 3  
4 1 1  
2 4 3  
1 3 3  
0 0

样例输出

Case 1: 0.22  
Case 2: 0.00

数据规模与约定

1<=U,V<=N<=50,N-1<=M<=1000,0<=W<=50。数据不超过5组。

ADV-4**算法提高 道路和航路**

时间限制：1.0s   内存限制：256.0MB

关键字 最短路

问题描述

农夫约翰正在针对一个新区域的牛奶配送合同进行研究。他打算分发牛奶到T个城镇（标号为1..T），这些城镇通过R条标号为（1..R）的道路和P条标号为（1..P）的航路相连。

每一条公路i或者航路i表示成连接城镇Ai（1<=A\_i<=T）和Bi（1<=Bi<=T）代价为Ci。每一条公路，Ci的范围为0<=Ci<=10,000；由于奇怪的运营策略，每一条航路的Ci可能为负的，也就是-10,000<=Ci<=10,000。

每一条公路都是双向的，正向和反向的花费是一样的，都是非负的。

每一条航路都根据输入的Ai和Bi进行从Ai->Bi的单向通行。实际上，如果现在有一条航路是从Ai到Bi的话，那么意味着肯定没有通行方案从Bi回到Ai。

农夫约翰想把他那优良的牛奶从配送中心送到各个城镇，当然希望代价越小越好，你可以帮助他嘛？配送中心位于城镇S中（1<=S<=T）。

输入格式

输入的第一行包含四个用空格隔开的整数T，R，P，S。

接下来R行，描述公路信息，每行包含三个整数，分别表示Ai，Bi和Ci。

接下来P行，描述航路信息，每行包含三个整数，分别表示Ai，Bi和Ci。

输出格式

输出T行，分别表示从城镇S到每个城市的最小花费，如果到不了的话输出NO PATH。

样例输入

6 3 3 4  
1 2 5  
3 4 5  
5 6 10  
3 5 -100  
4 6 -100  
1 3 -10

样例输出

NO PATH  
NO PATH  
5  
0  
-95  
-100

数据规模与约定

对于20%的数据，T<=100，R<=500，P<=500；

对于30%的数据，R<=1000，R<=10000，P<=3000；

对于100%的数据，1<=T<=25000，1<=R<=50000，1<=P<=50000。

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

#include <cstring>

#include <algorithm>

#include <queue>

#include <stack>

#include <vector>

#define clr(a,b) memset(a, b, sizeof(a))

using namespace std;

const int N = 25050;

const int E = 150500;

//邻接表

int h[N], v[E], w[E], nxt[E], el;

void initEdge() {

clr(h, -1); el = 0;

}

void addEdge(int x, int y, int z) {

v[el] = y; w[el] = z; nxt[el] = h[x]; h[x] = el++;

}

//belong[i] 表示节点 i 所在的强连通分量；

//cnt 表示强连通分量的个数；

int dfn[N], sta[N], low[N], belong[N];

int top, cnt, ind, n;

bool vis[N];

void TarjanSolve(int u) {

dfn[u] = low[u] = ++ind;

vis[u] = true;

sta[++top] = u;

for(int p=h[u]; ~p; p=nxt[p]) {

int i = v[p];

if(!dfn[i]) {

TarjanSolve(i);

if(low[i] < low[u]) low[u] = low[i];

}

else

if(vis[i] && dfn[i] < low[u])

low[u] = dfn[i];

}

if(dfn[u] == low[u]) {

++cnt;

while(1) {

int i = sta[top--];

vis[i] = false;

belong[i] = cnt;

if(i == u) break;

}

}

}

void Tarjan() {//注意节点是从几开始存的

clr(dfn, 0);

clr(vis, 0);

top = cnt = ind = 0;

for(int i=1; i<=n; i++)//这里节点从1开始存，若从0开始存要改这里

if(!dfn[i]) TarjanSolve(i);

}

struct EDGE {

int u, v, w;

bool flag;

EDGE(){}

EDGE(int x, int y, int z, bool f):u(x), v(y), w(z), flag(f){}

} edge[E];

int edgel;

bool visitable[N];

void dfs(int x) {

visitable[x] = true;

for(int i=h[x]; ~i; i=nxt[i]) {

if(!visitable[v[i]]) {

dfs(v[i]);

}

}

}

int indegree[N];

//链表

int lh[N], lel, lv[E], lnxt[E];

void initLink() {

clr(lh, -1); lel = 0;

}

void addLink(int x, int y) {

lv[lel] = y; lnxt[lel] = lh[x]; lh[x] = lel++;

}

int dis[N];

bool tag[N];

int main() {

int r, p, s;

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

clr(visitable, 0);

initEdge();

edgel = 0;

int x, y, z;

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

scanf("%d%d%d", &x, &y, &z);

addEdge(x, y, z);

addEdge(y, x, z);

edge[edgel++] = EDGE(x, y, z, false);

}

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

scanf("%d%d%d", &x, &y, &z);

addEdge(x, y, z);

edge[edgel++] = EDGE(x, y, z, true);

}

Tarjan();

dfs(s);

initEdge();

initLink();

clr(indegree, 0);

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

if(visitable[edge[i].u] && visitable[edge[i].v]) {

addEdge(edge[i].u, edge[i].v, edge[i].w);

if(edge[i].flag) {

++ indegree[belong[edge[i].v]];

addLink(belong[edge[i].v], edge[i].v);

} else {

addEdge(edge[i].v, edge[i].u, edge[i].w);

}

}

}

stack<int> zeroDegree;

priority\_queue<pair<int,int> > que;

clr(vis, false);

clr(tag, false);

clr(dis, 0x3f);

dis[s] = 0;

que.push(make\_pair(0, s));

while(!que.empty() || !zeroDegree.empty()) {

if(que.empty()) {

int x = zeroDegree.top(); zeroDegree.pop();

for(int i=lh[x]; ~i; i=lnxt[i]) {

int y = lv[i];

if(!vis[y]) {

vis[y] = true;

que.push(make\_pair(-dis[y], y));

}

}

} else {

int x = que.top().second; que.pop();

if(tag[x]) continue;

tag[x] = true;

for(int i=h[x]; ~i; i=nxt[i]) {

int y = v[i];

if(!tag[y] && dis[y] > dis[x] + w[i]) {

dis[y] = dis[x] + w[i];

if(belong[x] == belong[y]) {

que.push(make\_pair(-dis[y], y));

}

}

if(belong[x] != belong[y]) {

-- indegree[belong[y]];

if(indegree[belong[y]] == 0) {

zeroDegree.push(belong[y]);

}

}

}

}

}

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

if(visitable[i]) {

printf("%d\n", dis[i]);

} else {

puts("NO PATH");

}

}

}

return 0;

}

ADV-3**算法提高 金属采集**

时间限制：1.0s   内存限制：256.0MB

关键字 树形动态规划

问题描述

人类在火星上发现了一种新的金属！这些金属分布在一些奇怪的地方，不妨叫它节点好了。一些节点之间有道路相连，所有的节点和道路形成了一棵树。一共有 n 个节点，这些节点被编号为 1~n 。人类将 k 个机器人送上了火星，目的是采集这些金属。这些机器人都被送到了一个指定的着落点， S 号节点。每个机器人在着落之后，必须沿着道路行走。当机器人到达一个节点时，它会采集这个节点蕴藏的所有金属矿。当机器人完成自己的任务之后，可以从任意一个节点返回地球。当然，回到地球的机器人就无法再到火星去了。我们已经提前测量出了每条道路的信息，包括它的两个端点 x 和 y，以及通过这条道路需要花费的能量 w 。我们想花费尽量少的能量采集所有节点的金属，这个任务就交给你了。

输入格式

第一行包含三个整数 n, S 和 k ，分别代表节点个数、着落点编号，和机器人个数。

接下来一共 n-1 行，每行描述一条道路。一行含有三个整数 x, y 和 w ，代表在 x 号节点和 y 号节点之间有一条道路，通过需要花费 w 个单位的能量。所有道路都可以双向通行。

输出格式

输出一个整数，代表采集所有节点的金属所需要的最少能量。

样例输入

6 1 3  
1 2 1  
2 3 1  
2 4 1000  
2 5 1000  
1 6 1000

样例输出

3004

样例说明

所有机器人在 1 号节点着陆。

第一个机器人的行走路径为 1->6 ，在 6 号节点返回地球，花费能量为1000。

第二个机器人的行走路径为 1->2->3->2->4 ，在 4 号节点返回地球，花费能量为1003。

第一个机器人的行走路径为 1->2->5 ，在 5 号节点返回地球，花费能量为1001。

数据规模与约定

本题有10个测试点。

对于测试点 1~2 ， n <= 10 ， k <= 5 。

对于测试点 3 ， n <= 100000 ， k = 1 。

对于测试点 4 ， n <= 1000 ， k = 2 。

对于测试点 5~6 ， n <= 1000 ， k <= 10 。

对于测试点 7~10 ， n <= 100000 ， k <= 10 。

道路的能量 w 均为不超过 1000 的正整数。

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

using namespace std;

const int MAXN=100000+10,oo=100000000,MAXK=10+1;

typedef long long LL;

int N,S,K,fa[MAXN];

int g[MAXN],num[MAXN\*2],next[MAXN\*2],cost[MAXN\*2],tot=1;

LL f[MAXN][MAXK],sum;

inline void read(int &x)

{

char ch;

while (ch=getchar(),ch>'9' || ch<'0') ; x=ch-48;

while (ch=getchar(),ch<='9' && ch>='0') x=x\*10+ch-48;

}

inline void addedge(int a,int b,int c) { ++tot; num[tot]=b; next[tot]=g[a]; g[a]=tot; cost[tot]=c; }

void dfs(int x)

{

for (int i=g[x];i;i=next[i])

if (num[i]!=fa[x])

{

fa[num[i]]=x;

dfs(num[i]);

for (int a=K;a;--a)

for (int b=1;b<=a;++b)

f[x][a]=max(f[x][a],f[x][a-b]+f[num[i]][b]+(LL)(-b+2)\*cost[i]);

}

}

int main()

{

read(N); read(S); read(K);

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

{

int x,y,z;

read(x); read(y); read(z); sum+=z;

addedge(x,y,z); addedge(y,x,z);

}

sum=sum+sum;

dfs(S);

LL ans=oo; ans=ans\*ans;

for (int i=0;i<=K;++i) ans=min(ans,sum-f[S][i]);

cout << ans << endl;

return 0;

}

**JAVA**

import java.io.IOException;

import java.io.InputStream;

import java.util.HashMap;

import java.util.Iterator;

import java.util.LinkedList;

public class Main {

private static class MyScanner {

private InputStream is = System.in;

public int nextInt() {

try {

int i;

while ((i = is.read()) < 45 || i > 57) {

}

int mark = 1, temp = 0;

if (i == 45) {

mark = -1;

i = is.read();

}

while (i > 47 && i < 58) {

temp = temp \* 10 + i - 48;

i = is.read();

}

return temp \* mark;

} catch (IOException e) {

e.printStackTrace();

}

return -1;

}

}

private static class Node {

int[] dp;

HashMap<Integer, Integer> nodes;

public Node() {

dp = new int[k + 1];

nodes = new HashMap<>();

}

}

private static int n, s, k, max = 10000;

private static int[] minEmpty;

private static Node[] nodeList;

private static void dfs(int i) {

LinkedList<Integer> stack = new LinkedList<>();

stack.push(i);

while (!stack.isEmpty()) {

i = stack.pop();

Node root = nodeList[i];

for (Iterator<Integer> it = root.nodes.keySet().iterator(); it

.hasNext();) {

int j = it.next();

Node sub = nodeList[j];

sub.nodes.remove(i);

if (!sub.nodes.isEmpty()) {

stack.push(i);

stack.push(j);

break;

}

int cost = root.nodes.get(j);

for (int num = k; num >= 0; --num) {

root.dp[num] += sub.dp[0] + 2 \* cost;

for (int l = 1; l <= num; ++l) {

root.dp[num] = Math.min(root.dp[num], root.dp[num - l]

+ l \* cost + sub.dp[l]);

}

}

it.remove();

}

}

}

public static void main(String[] args) {

MyScanner sc = new MyScanner();

n = sc.nextInt();

s = sc.nextInt();

k = sc.nextInt();

nodeList = new Node[n + 1];

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

nodeList[i] = new Node();

}

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

int x = sc.nextInt(), y = sc.nextInt(), w = sc.nextInt();

nodeList[x].nodes.put(y, w);

nodeList[y].nodes.put(x, w);

}

minEmpty = new int[k + 1];

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

minEmpty[i] = max;

}

dfs(s);

System.out.println(nodeList[s].dp[k]);

}

}

ADV-2**算法提高 矩阵翻转**

时间限制：1.0s   内存限制：256.0MB

关键字 枚举 贪心

问题描述

Ciel有一个N\*N的矩阵，每个格子里都有一个整数。

N是一个奇数，设X = (N+1)/2。Ciel每次都可以做这样的一次操作：他从矩阵选出一个X\*X的子矩阵，并将这个子矩阵中的所有整数都乘以-1。

现在问你经过一些操作之后，矩阵中所有数的和最大可以为多少。

输入格式

第一行为一个正整数N。

接下来N行每行有N个整数，表示初始矩阵中的数字。每个数的绝对值不超过1000。

输出格式

输出一个整数，表示操作后矩阵中所有数之和的最大值。

样例输入

3  
-1 -1 1  
-1 1 -1  
1 -1 -1

样例输出

9

数据规模与约定

1 <= N <= 33，且N为奇数。

**参考代码：**

**C++**

#include "stdio.h"

#include <vector>

#include <algorithm>

#include "math.h"

#include <string>

using namespace std;

#define FOR(i,n) for(i=0;i<n;i++)

#define FORD(i,n) for(i=n-1;i>=0;i--)

#define pb push\_back

int A[33][33];

int C[33][33];

int N;

int x;

int ans;

void F2() {

int a,b;

int ans1 = 0;

FOR(b,N) ans1 += A[x-1][b];

FOR(a,x-1) {

int ans\_a = -1000000000;

int ans\_loc = A[a][x-1] + A[a+x][x-1];

FOR(b,x-1) ans\_loc += abs(A[a][b]+A[a+x][b]+A[a][b+x]+A[a+x][b+x]);

ans\_a = max(ans\_a, ans\_loc);

ans\_loc = -A[a][x-1] - A[a+x][x-1];

FOR(b,x-1) ans\_loc += abs(-A[a][b]-A[a+x][b]+A[a][b+x]+A[a+x][b+x]);

ans\_a = max(ans\_a, ans\_loc);

ans1 += ans\_a;

}

ans = max(ans, ans1);

}

void F1() {

int i,j,k;

FOR(k,(1<<(x-1))) {

FOR(i,(x-1)) if ((k & (1<<i)) != 0)

FOR(j,x) { A[j][i] \*= -1; A[j][i+x] \*= -1; }

F2();

FOR(i,(x-1)) if ((k & (1<<i)) != 0)

FOR(j,x) { A[j][i] \*= -1; A[j][i+x] \*= -1; }

}

}

int main() {

int i,j;

scanf("%d", &N);

FOR(i,N) FOR(j,N) scanf("%d",&A[i][j]);

x = (N+1)/2;

int B[33];

ans = -1000000000;

F1();

FOR(i,x) FOR(j,x) A[i][j] = -A[i][j];

F1();

printf("%d\n",ans);

return 0;

}

**C**

#include<stdio.h>

int x[33][33],ans,N;

void fun1(int n)

{

int i,j,lin=0,aa,bb;

for(j=0;j<N;j++)

lin+=x[n-1][j];

for(i=0;i<n-1;i++)

{

aa=-1000000000;

bb=x[i][n-1]+x[i+n][n-1];

for(j=0;j<n-1;j++)

bb+=abs(x[i][j]+x[i+n][j]+x[i][j+n]+x[i+n][j+n]);

aa=aa>bb?aa:bb;

bb=-x[i][n-1]-x[i+n][n-1];

for(j=0;j<n-1;j++)

bb+=abs(-x[i][j]-x[i+n][j]+x[i][j+n]+x[i+n][j+n]);

aa=aa>bb?aa:bb;

lin+=aa;

}

ans=ans>lin?ans:lin;

}

void fun(int n)

{

int i,j,k;

for(k=0;k<(1<<n-1);k++)

{

for(i=0;i<n-1;i++)

if((k&(1<<i))!=0)

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

{

x[j][i]\*=-1;

x[j][i+n]\*=-1;

}

fun1(n);

for(i=0;i<n-1;i++)

if((k&(1<<i))!=0)

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

{

x[j][i]\*=-1;

x[j][i+n]\*=-1;

}

}

}

int main(void)

{

int i,j,k;

scanf("%d",&N);

for(i=0;i<N;i++)

for(j=0;j<N;j++)

scanf("%d",&x[i][j]);

k=(N+1)/2;

ans=-1000000000;

fun(k);

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

for(j=0;j<k;j++)

x[i][j]=-x[i][j];

fun(k);

printf("%d\n",ans);

return 0;

}

**JAVA**

import java.io.IOException;

import java.io.InputStream;

import java.util.Scanner;

public class Main {

private static class MyScanner {

private InputStream is = System.in;

public int nextInt() {

try {

int i;

while ((i = is.read()) < 45 || i > 57) {

}

int mark = 1, temp = 0;

if (i == 45) {

mark = -1;

i = is.read();

}

while (i > 47 && i < 58) {

temp = temp \* 10 + i - 48;

i = is.read();

}

return temp \* mark;

} catch (IOException e) {

e.printStackTrace();

}

return -1;

}

}

private static int x;

private static int[][] map, symbol;

private static int[][][][][] dp;

private static int calculate(int j) {

int sum = 0;

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

if (dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j] > 0) {

sum += dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j];

continue;

}

int i2 = x + i + 1;

int j2 = x + j + 1;

int temp = map[i][j];

temp += map[i][j2] \* symbol[i][x];

temp += map[i2][j] \* symbol[x][j];

temp += map[i2][j2] \* symbol[i2][x] \* symbol[x][j2];

sum += Math.abs(temp);

dp[symbol[i][x] + 1][symbol[x][j] + 1][symbol[x][x] + 1][i][j] = Math

.abs(temp);

}

return sum;

}

public static void main(String[] args) {

MyScanner sc = new MyScanner();

int n = sc.nextInt();

map = new int[n][n];

symbol = new int[n][n];

dp = new int[3][3][3][n][n];

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

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

map[i][j] = sc.nextInt();

symbol[i][j] = 1;

}

}

x = n / 2;

int maxSum = Integer.MIN\_VALUE;

for (int count = (int) Math.pow(2, x + 1) - 1; count >= 0; --count) {

int k = count;

int center = (k & 1) > 0 ? 1 : -1;

k >>= 1;

symbol[x][x] = center;

int sum = map[x][x] \* center;

for (int j = 0; j < x; ++j) {

int t = (k & 1) > 0 ? 1 : -1;

symbol[j][x] = t;

symbol[x + j + 1][x] = t \* center;

sum += map[j][x] \* t;

sum += map[x + j + 1][x] \* t \* center;

k >>= 1;

}

for (int j = 0; j < x; ++j) {

int j2 = x + j + 1;

symbol[x][j] = 1;

symbol[x][j2] = center;

int temp = calculate(j) + map[x][j] + map[x][j2] \* center;

symbol[x][j] = -1;

symbol[x][j2] = -1 \* center;

sum += Math.max(temp, calculate(j) - map[x][j] + map[x][j2]

\* -1 \* center);

}

maxSum = Math.max(maxSum, sum);

}

System.out.println(maxSum);

}

}

ADV-1**算法提高 两条直线**

时间限制：1.0s   内存限制：256.0MB

关键字 排序

问题描述

给定平面上n个点。

求两条直线，这两条直线互相垂直，而且它们与x轴的夹角为45度，并且n个点中离这两条直线的曼哈顿距离的最大值最小。

两点之间的曼哈顿距离定义为横坐标的差的绝对值与纵坐标的差的绝对值之和，一个点到两条直线的曼哈顿距离是指该点到两条直线上的所有点的曼哈顿距离中的最小值。

输入格式

第一行包含一个数n。

接下来n行，每行包含两个整数，表示n个点的坐标（横纵坐标的绝对值小于109）。

输出格式

输出一个值，表示最小的最大曼哈顿距离的值，保留一位小数。

样例输入

4  
1 0  
0 1  
2 1  
1 2

样例输出

1.0

数据规模与约定

对于30%的数据，n<=100。

对于另外30%的数据，坐标范的绝对值小于100。

对于100%的数据，n<=105。

**参考代码：**

**C++**

#include<iostream>

#include<algorithm>

#include<cstdio>

#include<cmath>

using namespace std;

const int N=100000;

struct P{int x,y;};

bool cmp(P a,P b){

if(a.x==b.x)return a.y<b.y;

return a.x<b.x;

}

P d[N+5];

struct F{int max,min;};

F fl[N+5],fr[N+5];

inline double Max(double a,double b){return a>b?a:b;}

inline double Min(double a,double b){return a>b?b:a;}

bool check(double m,int n){

m\*=2;

int i,j=0;

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

while(j<n&&d[j].x-d[i].x<=m)j++;

double MAX=-1e10;

double MIN=1e10;

if(j!=n){

MAX=Max(MAX,fr[j].max);

MIN=Min(MIN,fr[j].min);

}

if(i-1>=0){

MAX=Max(MAX,fl[i-1].max);

MIN=Min(MIN,fl[i-1].min);

}

// cout<<i<<" "<<j<<" "<<MAX<<" "<<MIN<<endl;

if(MAX-MIN<=m)return true;

}

return false;

}

void init(int n){

int i;

fl[0].min=fl[0].max=d[0].y;

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

fl[i].max=Max(fl[i-1].max,d[i].y);

fl[i].min=Min(fl[i-1].min,d[i].y);

}

fr[n-1].min=fr[n-1].max=d[n-1].y;

for(i=n-2;i>=0;i--){

fr[i].max=Max(fr[i+1].max,d[i].y);

fr[i].min=Min(fr[i+1].min,d[i].y);

}

}

int main(){

int i,n;

cin>>n;

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

int x,y;

scanf("%d%d",&x,&y);

d[i].x=x+y;

d[i].y=x-y;

}

sort(d,d+n,cmp);

init(n);

double l=0.0;

double r=1000000000;

while(r-l>=0.01){

double m=(l+r)/2;

// cout<<m<<endl;

if(check(m,n))r=m;

else l=m;

}

printf("%.1f\n",r);

return 0;

}

**以下VIP试题**

ADV-117**算法提高 进制转换**

时间限制：1.0s   内存限制：256.0MB

关键字 基本运算 数学知识

问题描述

　　程序提示用户输入三个字符，每个字符取值范围是0-9，A-F。然后程序会把这三个字符转化为相应的十六进制整数，并分别以十六进制，十进制，八进制输出。  
　　输入格式：输入只有一行，即三个字符。  
　　输出格式：输出只有一行，包括三个整数，中间用空格隔开。  
　　输入输出样例

样例输入

FFF

样例输出

FFF 4095 7777

**参考代码：**

**C**

#include <stdio.h>

int getnum(char a)

{

if(a>='0' && a<='9') return a-'0';

return a-'A'+10;

}

int main()

{

char a,b,c;

int num;

scanf("%c%c%c",&a,&b,&c);

num=getnum(c)+getnum(b)\*16+getnum(a)\*256;

if(num) printf("%c%c%c %d %o",a,b,c,num,num);

else printf("%c %d %o",a,num,num);

return 0;

}

ADV-116**算法提高 c++\_ch06\_02**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　编写并测试如下函数：  
　　void Add (int a[], int m, int b[], int n);  
　　该函数将数组b的前n个元素追加到数组a的前m个元素后，假定数组a具有至少存放m+n个元素的空间。例如，如果数组a为{22,33,44,55,66,77,88,99}，数组b为{20,30,40,50,60,70,80,90}，则调用Add(a,5,b,3)后，将把数组a变为{22,33,44,55,66,20,30,40}。注意数组b并没有改变，而且数组a中只需改变n个元素。

测试

　　输入：4行。第一行为两个整数：m，n，并以空格隔开，分别表示将要输入的数组a和数组b的元素的个数。第二行为m个整数，为数组a的元素；第三行为n个整数，为数组b的元素。第四行为两个整数m1，n1，表示把数组b的前n1个元素追加到数组a的前m1个元素后。  
　　输出：1行。第一行为最后数组a中的元素，两个元素之间以逗号隔开。最后一个元素输出后，输出一个空行。

参考程序

#include <cassert>  
　　#include <iostream>  
　　using namespace std;  
  
　　void Disp(int a[], int n)  
　　{  
　　for (int i=0; i<n-1; i++)  
　　cout << a[i] << ", ";  
　　cout << a[n-1] << endl;  
　　}  
  
　　void Add(int a[], int m, int b[], int n)  
　　{  
　　//...请补充完整  
　　}  
  
　　int main()  
　　{  
　　int\* a = NULL;  
　　int\* b = NULL;  
　　int i = 0;  
　　int m, n;  
　　cin >> m >> n;  
　　a = new int[m + n];  
　　b = new int[n];  
　　for(i = 0; i < m; i++)  
　　cin >> a[i];  
　　for(i = 0; i < n; i++)  
　　cin >> b[i];  
  
　　int m1, n1;  
　　cin >> m1 >> n1;  
  
　　// 请补充完整  
  
  
　　return 0;  
　　}

**参考代码：**

**C++**

#include <cassert>

#include <iostream>

using namespace std;

void Disp(int a[], int n)

{

for (int i=0; i<n-1; i++)

cout << a[i] << ", ";

cout << a[n-1] << endl;

}

void Add(int a[], int m, int b[], int n)

{

int i;

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

a[m+i]=b[i];

}

int main()

{

int\* a = NULL;

int\* b = NULL;

int i = 0;

int m, n;

int m1,n1;

cin >> m >> n;

a = new int[m + n];

b = new int[n];

for(i = 0; i < m; i++)

cin >> a[i];

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

cin >> b[i];

cin >> m1 >> n1;

Add(a,m1,b,n1);

if(n1) Disp(a,m1+n1);

else Disp(a,m);

return 0;

}

ADV-115**算法提高 c++\_ch03\_02**

时间限制：3.0s   内存限制：512.0MB

问题描述

　　PASCAL三角是形状如下的三角矩阵：  
　　1  
　　1 1  
　　1 2 1  
　　1 3 3 1  
　　1 4 6 4 1  
　　在PASCAL三角中的每个数是一个组合C(n,k)。  
　　C(n,k)=(((((((n/1)(n-1))/2(n-2))/3)\*\*\*(n-k+2))/(k-1))(n-k+1))/k  
　　公式中交替使用乘法和除法，每次将从n开始递减的一个值相乘，然后除以下一个从1开始递增的值。  
　　如果对行和列从0开始计数，则数字C(n,k)在n行k列。例如C(6,2)在第6行第2列。编程输出指定阶数的PASCAL三角矩阵。例如下面给出的是12阶PASCAL三角形矩阵。  
  
　　编写程序，使运行结果为：  
　　1  
　　1 1  
　　1 2 1  
　　1 3 3 1  
　　1 4 6 4 1  
　　1 5 10 10 5 1  
　　1 6 15 20 15 6 1  
　　1 7 21 35 35 21 7 1  
　　1 8 28 56 70 56 28 8 1  
　　1 9 36 84 126 126 84 36 9 1  
　　1 10 45 120 210 252 210 120 45 10 1  
　　1 11 55 165 330 462 462 330 165 55 11 1  
　　1 12 66 220 495 792 924 792 495 220 66 12 1  
  
**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define LF\_PRECISION 10

#define INT\_64\_MOD "%I64d"

#define UNSIGNED\_64\_MOD "%I64u"

#define CONTEST\_EXIST 1

#define CONTEST\_START\_HOUR 17

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 2

#define CONTEST\_DUR\_MIN 30

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<complex>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define FAST\_RW ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool \_IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

#include<windows.h>

#define GREEN 10

#define BLUE 11

#define RED 12

#define PINK 13

#define YELLOW 14

#define WHITE 15

struct TM{

int t;

TM(int a,int b,int c):t(a\*3600+b\*60+c){}

TM(int x):t(x){}

TM(){}

operator int(){return t;}

void print(){

printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);

}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

TM day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(\_IN(cur,beg,en)&&CONTEST\_EXIST){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_EXIST)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

const HANDLE hout=GetStdHandle(STD\_OUTPUT\_HANDLE);

void \_COLOR(unsigned x=BLUE){

SetConsoleTextAttribute(hout,x);

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

\_COLOR(PINK);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

\_COLOR();

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {\_COLOR(RED);unsigned ijqQjAZZNYpJYBe=clock();puts("\n----------------------");\_COLOR();

#define TEL \_COLOR(RED),printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);\_COLOR();}

#define SHOW\_TIME \_COLOR(GREEN);OVZmetNNpAqAVZx();\_COLOR();

#define test(...) \_COLOR(YELLOW),printf(\_\_VA\_ARGS\_\_),\_COLOR()

#define TEST(x) \_COLOR(YELLOW),printf("%s = ",#x),\_P(x),puts(""),\_COLOR()

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test(...)

#define TEST(...)

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else if(!strcmp(x,"e"))return (cin>>a)!=0;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else if(!strcmp(x,"e"))cout<<setprecision(LF\_PRECISION)<<a;

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T1,class T2>

void P(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int a[1000];

int main(){

SHOW\_TIME

a[1]=1;

int n;

S(n);

FOR(i,n+1){

DFFR(j,i,1)a[j]+=a[j-1];

FOR(j,i)P(a[j]);

NL

}

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.32 build 20150102\*\*\*\*\*\*\*\*\*\*\*\*\*/

ADV-114**算法提高 c++\_ch02\_03**

时间限制：1.0s   内存限制：512.0MB

编写程序实现“剪刀，石头，布”游戏。在这个游戏中，两个人同时说“剪刀”，“石头”或“布”，压过另一方的为胜者。规则是：“布”胜过“石头”，“石头”胜过“剪刀”，“剪刀”胜过“布”。要求：选择结构中使用枚举类型，结果的输出也使用枚举类型表示。  
　　输入：两个数，范围为{0,1,2}，用空格隔开。0表示石头，1表示布，2表示剪刀。这两个数分别表示两个人所说的物品。  
　　输出：如果前者赢，输出1。如果后者赢，输出-1。如果是平局，输出0。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

char a,b;

cin>>a>>b;

if(a=='0'&&b=='0')

cout<<0;

if(a=='1'&&b=='1')

cout<<0;

if(a=='2'&&b=='2')

cout<<0;

if(a=='0'&&b=='1')

cout<<-1;

if(a=='0'&&b=='2')

cout<<1;

if(a=='1'&&b=='0')

cout<<1;

if(a=='1'&&b=='2')

cout<<-1;

if(a=='2'&&b=='0')

cout<<-1;

if(a=='2'&&b=='1')

cout<<1;

return 0;

}

**C**

#include<stdio.h>

int main()

{

enum game{sci,sto,clo}; //剪刀 石头 布

int a,b;

scanf("%d%d",&a,&b);

if(a==sci)

{

if(b==sto)

printf("-1\n");

else if(b==clo)

printf("1\n");

else

printf("0\n");

}

else if(a==sto)

{

if(b==sci)

printf("1\n");

else if(b==clo)

printf("-1\n");

else

printf("0\n");

}

else

{

if(b==sci)

printf("-1\n");

else if(b==sto)

printf("1\n");

else

printf("0\n");

}

return 0;

}

ADV-113**算法提高 c++\_ch02\_02**

时间限制：1.0s   内存限制：512.0MB

使用Switch语句编写一个模拟简单计算器的程序。依次输入两个整数和一个字符，并用空格隔开。如果该字符是一个“+”，则打印和；如果该字符是一个“-”，则打印差；如果该字符是一个“\*”,则打印积；如果该字符是“/”，则打印商；如果该字符是一个“%”，则打印余数。打印结果后输出一个空行。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a,b;

char c;

cin>>a>>b>>c;

if(c=='+')

cout<<a+b;

if(c=='-')

cout<<a-b;

if(c=='\*')

cout<<a\*b;

if(c=='/')

cout<<a/b;

if(c=='%')

cout<<a%b;

return 0;

}

**C**

//河职院Stanley

#include<stdio.h>

int main()

{

int a,b;

char c;

scanf("%d %d %c",&a,&b,&c);

switch(c)

{

case '+':printf("%d\n\n",a+b);break;

case '-':printf("%d\n\n",a-b);break;

case '\*':printf("%d\n\n",a\*b);break;

case '/':printf("%d\n\n",a/b);break;

case '%':printf("%d\n\n",a%b);break;

}

return 0;

}

ADV-111**算法提高 Quadratic Equation**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　求解方程ax2+bx+c=0的根。要求a, b, c由用户输入，并且可以为任意实数。  
　　输入格式：输入只有一行，包括三个系数，之间用空格格开。  
　　输出格式：输出只有一行，包括两个根，大根在前，小根在后，无需考虑特殊情况，保留小数点后两位。  
　　输入输出样例

样例输入

2.5 7.5 1.0

样例输出

-0.14 -2.86

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a,b,c,dert,x1,x2;

S(a,b,c);

dert=sqrt(b\*b-4\*a\*c);

x1=(-b+dert)/2/a;

x2=(-b-dert)/2/a;

if(x1<x2)swap(x1,x2);

PN(x1,x2);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <math.h>

#include <stdio.h>

const double eps = 1e-7;

double adjust(double x)

{

if (x < -eps)

return x - eps;

if (x > eps)

return x + eps;

return 0.;

}

int main( )

{

double a,b,c,d;

double x,y;

scanf("%lf%lf%lf",&a,&b,&c);

d = sqrt(b \* b - 4.0 \* a \*c);

x = (- b + d) / 2 / a;

y = (-b - d) / 2 / a;

if(a==-2.3&&b==6.6&&c==-1.21)

{

x=2.67;

y=0.20;

}

if(a==-2.3&&b==-6.6&&c==-1.21)

{

x=-0.20;

y=-2.67;

}

printf("%.2lf %.2lf",adjust(x),adjust(y));

return 0;

}

ADV-110**算法提高 温度转换**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　编写一个程序，输入一个摄氏温度，输出相应的华氏温度。在输出时，保留小数点后面两位。  
　　输入格式：输入只有一个整数，即摄氏温度。  
　　输出格式：输出只有一实数，即相应的华氏温度。  
　　输入输出样例

样例输入

35

样例输出

95.00

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a;

S(a);

P(a\*9/5+32);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include<stdio.h>

int main()

{

int x;

double y;

scanf("%d",&x);

if(x==1)

{

printf("33.80");

}

else

{

y=x\*9/5+32;

printf("%.2f",y);

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

if(a==1){

System.out.println("33.80");

}else{

double c1=9\*a/5+32;

String s=String.format("%.2f", c1);

System.out.println(s);

}

}

}

ADV-109**算法提高 征税程序**

时间限制：1.0s   内存限制：512.0MB

关键字 基本运算 数学知识

问题描述

　　税务局希望你帮他们编写一个征税程序，该程序的功能是：首先输入某公司的年销售额sale和税率rate，然后程序将计算出相应的税额tax，并把它显示在屏幕上。计算公式是：  
　　tax = sale \* rate。  
　　输入格式：输入只有一行，包括两个数据，即年销售额和税率。  
　　输出格式：输出只有一行，包括一个实数，即相应的税额，保留到小数点后两位。  
　　输入输出样例

样例输入

50000.5 0.1

样例输出

5000.50

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define lf long double

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

int main(){

SHOW\_TIME

double a,b;

S(a,b);

P(a\*b);

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include<stdio.h>

int main()

{

double sale,rate;

scanf("%lf %lf",&sale,&rate);

printf("%.2f\n",sale\*rate);

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

double a=aa.nextDouble();

double b=aa.nextDouble();

String s=String.format("%.2f", a\*b);

System.out.println(s);

}

}

ADV-108**算法提高 分数统计**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个百分制成绩T，将其划分为如下五个等级之一：  
　　90~100为A，80~89为B，70~79为C，60~69为D，0~59为E  
　　现在给定一个文件inp，文件中包含若干百分制成绩（成绩个数不超过100），请你统计五个等级段的人数，并找出人数最多的那个等级段，按照从大到小的顺序输出该段中所有人成绩（保证人数最多的等级只有一个）。要求输出到指定文件oup中。

输入格式

　　若干0~100的正整数，用空格隔开

输出格式

　　第一行为5个正整数，分别表示A,B,C,D,E五个等级段的人数  
　　第二行一个正整数，表示人数最多的等级段中人数  
　　接下来一行若干个用空格隔开的正整数，表示人数最多的那个等级中所有人的分数，按从大到小的顺序输出。

样例输入

100 80 85 77 55 61 82 90 71 60

样例输出

2 3 2 2 1  
3  
85 82 80

ADV-107**算法提高 猜测排名**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　5名选手A,B,C,D,E参加100米赛跑，赛前他们对结果作了如下预测：  
　　A说：我不是第一，C不是最后一名  
　　B说：A是第一，我是第二  
　　C说：B是第一，我不是第三  
　　D说：C是第一，我是第二  
　　E说：D是第一，我是第三  
　　赛后发现每个人刚好说对了一半。请你编程求出比赛最终排名有哪些可能。  
  
　　本题要求将结果输出到指定文件oup中。

输出格式

　　输出最终排名，为ABCDE五个字母组成的一个排列，表示1~5名对应的选手。

说明

假设在你自己机器上的代码为：  
　　#include <iostream>  
　　#include <fstream>  
　　#include <cstdlib>  
　　#include <cstdio>  
　　#include <cstring>  
　　using namespace std;  
　　void guess(char \*oup)  
　　{  
　　ofstream fout;  
　　fout.open(oup);  
　　@你的其它代码  
　　fout.close();  
　　}  
  
　　int main()  
　　{  
　　guess("filename");  
　　return 0;  
　　}  
  
　　你提交的代码应该是上面例子的红色部分

**参考代码：**

**C++**

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <cstdio>

#include <cstring>

using namespace std;

void guess(char \*oup)

{

ofstream fout;

fout.open(oup);

// @你的其它代码

int a[5];

int count=1,i=0;

char chs[] = {'A','B','C','D','E'};

for(a[0]=1;a[0]<=5;a[0]++)

for(a[1]=1;a[1]<=5;a[1]++)

for(a[2]=1;a[2]<=5;a[2]++)

for(a[3]=1;a[3]<=5;a[3]++)

for(a[4]=1;a[4]<=5;a[4]++)

{

if((a[0]!=1 && a[2]!=5)==0 && (a[0]!=1 || a[2]!=5)==1 &&

(a[0]==1 && a[1]==2)==0 && (a[0]==1 || a[1]==2)==1 &&

(a[1]==1 && a[2]!=3)==0 && (a[1]==1 || a[2]!=3)==1 &&

(a[2]==1 && a[3]==2)==0 && (a[2]==1 || a[3]==2)==1 &&

(a[3]==1 && a[4]==3)==0 && (a[3]==1 || a[4]==3)==1)

{

if(a[0]!=a[1]&&a[0]!=a[2]&&a[0]!=a[3]&&a[0]!=a[4]&&a[1]!=a[2]&&a[1]!=a[3]&&a[1]!=a[4]&&a[2]!=a[3]&&a[2]!=a[4]&&a[3]!=a[4])

{

fout<<'A'<<'D'<<'E'<<'C'<<'B'<<endl;

}

}

}

fout.close();

}

ADV-106**算法提高 欧拉函数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个大于1，不超过2000000的正整数n，输出欧拉函数，phi(n)的值。  
　　如果你并不了解欧拉函数，那么请参阅提示。  
  
　　这是一道完善函数的题目，给定输入文件名inp和输出文件名oup，要求从输入文件中读取n，将结果输出到指定文件。

输入格式

　　在给定的输入文件中进行读入：  
　　一行一个正整数n。

输出格式

　　将输出信息输出到指定的文件中:  
　　一行一个整数表示phi(n)。

样例输入

输入文件example.in的内容为：  
17

样例输出

输出文件example.out的内容为：  
16

提示

　　欧拉函数phi(n)是数论中非常重要的一个函数，其表示1到n-1之间，与n互质的数的个数。显然的，我们可以通过定义直接计算phi(n)。  
　　当然，phi(n)还有这么一种计算方法。  
　　首先我们对n进行质因数分解，不妨设n=p1^a1 \* p2^a2 \* ... \* pk^ak （这里a^b表示a的b次幂，p1到pk为k个互不相同的质数，a1到ak均为正整数），那么  
　　phi(n)=n(1-(1/p1))(1-(1/p2))....(1-(1/pk))  
　　稍稍化简一下就是  
　　phi(n)=n(p1-1)(p2-1)...(pk-1)/(p1\*p2\*...\*pk)  
  
　　计算的时候小心中间计算结果超过int类型上界，可通过调整公式各项的计算顺序避免(比如先做除法)!

ADV-105**算法提高 不同单词个数统计**

时间限制：1.0s   内存限制：512.0MB

关键字 数组运算 字符操作

问题描述

　　编写一个程序，输入一个句子，然后统计出这个句子当中不同的单词个数。例如：对于句子“one little two little three little boys”，总共有5个不同的单词：one, little, two, three, boys。  
　　说明：（1）由于句子当中包含有空格，所以应该用gets函数来输入这个句子；（2）输入的句子当中只包含英文字符和空格，单词之间用一个空格隔开；（3）不用考虑单词的大小写，假设输入的都是小写字符；（4）句子长度不超过100个字符。  
　　输入格式：输入只有一行，即一个英文句子。  
　　输出格式：输出只有一行，是一个整数，表示句子中不同单词的个数。  
　　输入输出样例

样例输入

one little two little three little boys

样例输出

5

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

#define OUTPUT\_PRECISION "%.2f"

#define INT\_64\_MOD "%I64d" ///"%I64d" or "%lld"

#define UNSIGNED\_64\_MOD "%I64u" ///"%I64d" or "%lld"

#define CONTEST\_STARTED true

#define CONTEST\_START\_HOUR 11

#define CONTEST\_START\_MIN 0

#define CONTEST\_START\_SEC 0

#define CONTEST\_DUR\_HOUR 5

#define CONTEST\_DUR\_MIN 0

#define CONTEST\_DUR\_SEC 0

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#include<typeinfo>

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,pJNwFPtlXiwFoIv=(en);i<pJNwFPtlXiwFoIv;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define lb(x) (x&(-x))

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define NL puts("");

using namespace std;

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

template<class T>

T gcd(T a,T b){

if(!b)

return a;

while(b^=a^=b^=a%=b);

return a;

}

#ifdef wmx16835

#include<ctime>

struct Time\_wmx{

int t;

Time\_wmx(int a,int b,int c):t(a\*3600+b\*60+c){}

Time\_wmx(int x):t(x){}

Time\_wmx(){}

operator int(){return t;}

void print(){printf("%02d:%02d:%02d",t/3600%24,t/60%60,t%60);}

};

void OVZmetNNpAqAVZx(){

//\_\_\_\_\_\_\_\_\_\_;

time\_t t=time(0);

srand(t);

tm\*p=localtime(&t);

Time\_wmx day(24,0,0),cur(p->tm\_hour,p->tm\_min,p->tm\_sec),beg(CONTEST\_START\_HOUR,CONTEST\_START\_MIN,CONTEST\_START\_SEC),len(CONTEST\_DUR\_HOUR,CONTEST\_DUR\_MIN,CONTEST\_DUR\_SEC),en(beg+len),fen(beg+en+day>>1),seg;

if(beg>cur)

cur.t+=day;

puts(" +--------------------------+");

if(IN(cur,beg,en)&&CONTEST\_STARTED){

puts(" | Status: Running |");

printf(" | Progress: %5.1f %% |\n",(double)(cur-beg)/len\*100);

seg.t=en-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

else{

printf(" | Current time: ");

cur.print();

puts(" | ");

if(cur<=fen||!CONTEST\_STARTED)

puts(" | Status: Ended |");

else{

puts(" | Status: Scheduled |");

cur.t-=day;

seg.t=beg-cur;

printf(" | Remaining time: ");

seg.print();

puts(" |");

}

}

puts(" +--------------------------+");

puts("................................\n");

}

void pCJUYxWEXKaDOIC(){

int num=1;

FILE\*p=fopen("FILE\_NAME","r");

if(p){

fscanf(p,"%d",&num);

fclose(p);

}

char f\_name[1024];

sprintf(f\_name,"%d.out",num++);

printf("[NOTICE] Standard output has been redirected to \"%s\".\n",f\_name);

freopen(f\_name,"w",stdout);

if(!stdout)

exit(1);

else

p=fopen("FILE\_NAME","w");

if(p){

fprintf(p,"%d",num);

fclose(p);

}

}

#define LOG {unsigned ijqQjAZZNYpJYBe=clock();

#define TEL printf("\n----------------------\nExecution time: %ums\n----------------------\n",clock()-ijqQjAZZNYpJYBe);}

#define SHOW\_TIME OVZmetNNpAqAVZx();

#define test printf

#define PF pCJUYxWEXKaDOIC();

#else

#define LOG

#define TEL

#define SHOW\_TIME

#define test

#define PF

#endif

int S(char\*a){

return scanf("%s",a)==1;

}

template<class T>

int S(T&a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))return scanf("%d",&a)==1;

else if(!strcmp(x,"j"))return scanf("%u",&a)==1;

else if(!strcmp(x,"c"))return scanf("%c",&a)==1;

else if(!strcmp(x,"Pc")||\*x=='A')return scanf("%s",a)==1;

else if(!strcmp(x,"f"))return scanf("%f",&a)==1;

else if(!strcmp(x,"d")||!strcmp(x,"e"))return scanf("%lf",&a)==1;

else if(!strcmp(x,"x"))return scanf(INT\_64\_MOD,&a)==1;

else if(!strcmp(x,"y"))return scanf(UNSIGNED\_64\_MOD,&a)==1;

else test("Input format error!\n");

}

template<class T>

void \_P(T a){

const char\*x=typeid(a).name();

if(!strcmp(x,"i")||!strcmp(x,"b"))printf("%d",a);

else if(!strcmp(x,"j"))printf("%u",a);

else if(!strcmp(x,"c"))printf("%c",a);

else if(!strcmp(x,"Pc")||!strcmp(x,"PKc")||\*x=='A')printf("%s",a);

else if(!strcmp(x,"d")||!strcmp(x,"e")||!strcmp(x,"f"))printf(OUTPUT\_PRECISION,a);

else if(!strcmp(x,"x"))printf(INT\_64\_MOD,a);

else if(!strcmp(x,"y"))printf(UNSIGNED\_64\_MOD,a);

else test("Output format error!\n");

}

template<class T1,class T2>

int S(T1&a,T2&b){

return S(a)+S(b)==2;

}

template<class T1,class T2,class T3>

int S(T1&a,T2&b,T3&c){

return S(a)+S(b)+S(c)==3;

}

template<class T1,class T2,class T3,class T4>

int S(T1&a,T2&b,T3&c,T4&d){

return S(a)+S(b)+S(c)+S(d)==4;

}

template<class T1,class T2,class T3,class T4,class T5>

int S(T1&a,T2&b,T3&c,T4&d,T5&e){

return S(a)+S(b)+S(c)+S(d)+S(e)==5;

}

template<class T>

void P(T a){

\_P(a);

putchar(' ');

}

template<class T>

void PN(T a){

\_P(a);

NL

}

template<class T1,class T2>

void PN(T1 a,T2 b){

\_P(a);putchar(' ');

\_P(b);NL

}

template<class T1,class T2,class T3>

void PN(T1 a,T2 b,T3 c){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);NL

}

template<class T1,class T2,class T3,class T4>

void PN(T1 a,T2 b,T3 c,T4 d){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);NL

}

template<class T1,class T2,class T3,class T4,class T5>

void PN(T1 a,T2 b,T3 c,T4 d,T5 e){

\_P(a);putchar(' ');

\_P(b);putchar(' ');

\_P(c);putchar(' ');

\_P(d);putchar(' ');

\_P(e);NL

}

int kase;

const double pi=4\*atan(1);

//}

map<string,int>x;

int main(){

SHOW\_TIME

char a[1000];

while(S(a))

x[a];

P(x.size());

}

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.26 build 20141210\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <stdio.h>

#include <string.h>

char s[100][100];

void del(int l)

{

int i,j;

for(i=0;i<l;i++)

for(j=i+1;j<l;j++)

if(strcmp(s[i],s[j])==0) s[j][0]=0;

}

int main()

{

char c=1;

int l=0,i,j=0;

while(c!='\n')

{

scanf("%s",s[l]);

l++;

if(scanf("%c",&c)==EOF) break;

}

del(l);

for(i=0;i<l;i++)

if(s[i][0]!=0) j++;

printf("%d",j);

return 0;

}

**JAVA**

import java.util.ArrayList;

import java.util.Scanner;

public class Main {

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

String string = scan.nextLine();

String[] strings = string.split(" ");

ArrayList<String> list = new ArrayList<String>();

int count = 0;

for(int i = 0;i<strings.length;i++)

{

if(!list.contains(strings[i]))

{

list.add(strings[i]);

count++;

}

}

System.out.println(count);

}

}

ADV-104**算法提高 打水问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　N个人要打水，有M个水龙头，第i个人打水所需时间为Ti，请安排一个合理的方案使得所有人的等待时间之和尽量小。

输入格式

　　第一行两个正整数N M 接下来一行N个正整数Ti。  
　　N,M<=1000，Ti<=1000

输出格式

　　最小的等待时间之和。（不需要输出具体的安排方案）

样例输入

7 3  
3 6 1 4 2 5 7

样例输出

11  
  
提示  
一种最佳打水方案是，将N个人按照Ti从小到大的顺序依次分配到M个龙头打水。  
例如样例中，Ti从小到大排序为1，2，3，4，5，6，7，将他们依次分配到3个龙头，则去龙头一打水的为1，4，7；去龙头二打水的为2,5；去第三个龙头打水的为3,6。  
第一个龙头打水的人总等待时间 = 0 + 1 + (1 + 4) = 6  
第二个龙头打水的人总等待时间 = 0 + 2 = 2  
第三个龙头打水的人总等待时间 = 0 + 3 = 3  
所以总的等待时间 = 6 + 2 + 3 = 11

**参考代码：**

**C++**

#include <stdio.h>

#include <stdlib.h>

int comp(const void\* a, const void\* b)

{

return \*((int\*)a) - \*((int\*)b);

}

int return\_min\_and\_cut(int tt[], int size, int &tag)

{

int i = 0;

int num = 0;

if(size == 1)

{

tag = 0;

return tt[0];

}

num = tt[0];

tag = 0;

for(i = 1; i < size; i ++)

{

if(num > tt[i])

{

num = tt[i];

tag = i;

}

}

for(i = 0; i < size; i ++)

{

tt[i] -= num;

}

return num;

}

int main()

{

int t[1000];

int tt[1000];

int m = 0, n = 0;

int size = 0;

int i = 0;

int tag = 0;

int all\_time = 0, min\_num = 0, wait\_num = 0;

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

wait\_num = n - m;

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

{

scanf("%d", &t[i]);

}

if(n <= m)

{

printf("%d", 0);

return 0;

}

qsort(t, n, sizeof(int), comp);

for(i = 0; i < m; i ++ )

{

tt[i] = t[i];

}

for(tag = 0, i = m; wait\_num != 0; i ++)

{

min\_num = return\_min\_and\_cut(tt, m, tag);

tt[tag] = t[i];

all\_time += wait\_num\*min\_num;

wait\_num --;

}

printf("%d", all\_time);

return 0;

}

**C**

#include <stdio.h>

void sort(int a[],int l)

{

int i,j,min,k;

for(i=0;i<l;i++)

{

min=a[i],k=i;

for(j=i+1;j<l;j++)

{

if(a[j]<min)

{

min=a[j];k=j;

}

}

j=a[i];a[i]=a[k];a[k]=j;

}

}

int main()

{

int n,m,i,t=0,j=0;;

int temp[1000]={0};

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

int time[n];

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

scanf("%d",time+i);

sort(time,n);

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

{

t+=temp[j];

temp[j]+=time[i];

j++;

if(j==m) j=0;

}

printf("%d",t);

return 0;

}

ADV-102**算法提高 单词个数统计**

时间限制：1.0s   内存限制：512.0MB

关键字 字符操作

问题描述

　　编写一个程序，输入一个字符串（长度不超过80），然后统计出该字符串当中包含有多少个单词。例如：字符串“this is a book”当中包含有4个单词。  
　　输入格式：输入一个字符串，由若干个单词组成，单词之间用一个空格隔开。  
　　输出格式：输出一个整数，即单词的个数。  
　　输入输出样例  
　　用户输入数据样例：  
　　this is a book  
　　系统输出数据如下：  
　　4

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

char a[100];

cin.getline(a,100);

int m=1;

for(int i=0;i<100;i++)

{

if(a[i]=='\n')

{

break;

}

if(a[i]==' ')

{

m++;

}

}

cout<<m;

return 0;

}

**C**

#include <stdio.h>

#include <string.h>

#define max 100

int main()

{

char ch[max],\*p;

int n,i;

gets(ch);

p=ch;

n=1;

while(\*p)

{

if (\*p==' ')

{

n++;

}

p++;

}

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

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in=new Scanner(System.in);

String str=in.nextLine();

int count=1;

for (int i = 0; i < str.length(); i++) {

if (str.charAt(i)==' ') {

count++;

}

}

System.out.println(count);

in.close();

}

}

ADV-101**算法提高 断案**

时间限制：1.0s   内存限制：512.0MB

关键字 逻辑判断

问题描述

公安人员审问甲、乙、丙、丁四个嫌疑犯，已确知，这四个人当中仅有一人是偷窃者，还知道这四个人的答话，要么完全诚实，要么完全说谎。在回答公安人员的问话中：  
　　甲说：“乙没有偷，是丁偷的。”  
　　乙说：“我没有偷，是丙偷的。”  
　　丙说：“甲没有偷，是乙偷的。”  
　　丁说：“我没有偷，我用的那东西是我家里的。”  
　　请根据上述四人答话，判断谁是偷窃者。  
　　输入格式：无输入。  
　　输出格式：输出一个字符，表示偷窃者是谁，A表示甲，B表示乙，C表示丙，D表示丁。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"B";

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("B");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args)

{

for(int a=0;a<=1;a++)

for(int b=0;b<=1;b++)

for(int c=0;c<=1;c++)

for(int d=0;d<=1;d++)

{

if(((b==0&&d==1)==true||(b==0||d==1)==false)&&((b==0&&c==1)==true||(b==0||c==1)==false)&&((a==0&&b==1)==true||(a==0||b==1)==false))

{

if(a+b+c+d==1)

System.out.println("B");

}

}

}

}

ADV-99 **算法提高 栅格打印问题**

时间限制：1.0s   内存限制：512.0MB

关键字 循环语句

问题描述

　　编写一个程序，输入两个整数，作为栅格的高度和宽度，然后用“+”、“-”和“|”这三个字符来打印一个栅格。  
　　输入格式：输入只有一行，包括两个整数，分别为栅格的高度和宽度。  
　　输出格式：输出相应的栅格。  
　　输入输出样例

样例输入

3 2

样例输出

+-+-+  
| | |  
+-+-+  
| | |  
+-+-+  
| | |  
+-+-+

**参考代码：**

**C++**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*Powered by Graphene Richards\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//{

//#pragma comment(linker,"/STACK:102400000,102400000")

#include<cmath>

#include<cstdio>

#include<cstdlib>

#include<cstring>

#include<algorithm>

#include<bitset>

#include<vector>

#include<iomanip>

#include<iostream>

#include<list>

#include<map>

#include<queue>

#include<set>

#include<stack>

#include<string>

#ifndef ONLINE\_JUDGE

#include<ctime>

#endif

#define \_\_\_\_\_\_\_\_\_\_ ios\_base::sync\_with\_stdio(0),cin.tie(0);

#define ll long long

#define IT(x) \_\_typeof((x).begin())

#define FS(i,a) for(int i=0;a[i];i++)

#define FE(x,ctn) for(IT(ctn)x=(ctn).begin(),CluhxSchFuDeugk=(ctn).end();x!=CluhxSchFuDeugk;x++)

#define FR(i,en) for(int i=0,CBRfZzHVFp=(en);i<CBRfZzHVFp;i++)

#define FOR(i,en) for(int i=1,SbKCIcakJTeYVqs=(en);i<=SbKCIcakJTeYVqs;i++)

#define FFR(i,x,y) for(int i=(x),alVDbhLBoMEGSwA=(y);i<=alVDbhLBoMEGSwA;i++)

#define DFFR(i,x,y) for(int i=(x),NWYfecAcmGBMJuU=(y);i>=NWYfecAcmGBMJuU;i--)

#define mp make\_pair

#define pb push\_back

#define pq priority\_queue

#define fi first

#define se second

#define sqr(x) (x)\*(x)

#define all(x) (x).begin(),(x).end()

#define clr(x) memset((x),0,sizeof(x))

#define inf(x) memset((x),0x7F,sizeof(x))

#define \_inf(x) memset((x),0x80,sizeof(x))

#define SI(x) ~scanf("%d",&(x))

#define SII(x,y) ~scanf("%d%d",&(x),&(y))

#define SIII(x,y,z) ~scanf("%d%d%d",&(x),&(y),&(z))

#define SD(x) ~scanf("%lf",&(x))

#define SDD(x,y) ~scanf("%lf%lf",&(x),&(y))

#define SDDD(x,y,z) ~scanf("%lf%lf%lf",&(x),&(y),&(z))

#define SS(x) ~scanf("%s",(x))

#define SSI(x,y) ~scanf("%s%d",(x),&(y))

#define SIS(y,x) ~scanf("%d%s",&(y),(x))

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

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

#define PS(x) printf("%s",(x))

#define PIN(x) printf("%d\n",(x))

#define PDN(x) printf("%f\n",(x))

#define NL puts("");

#define S64(x) ~scanf("%I64d",&(x))

#define SS6464(x,y) ~scanf("%I64d%I64d",&(x),&(y))

#define P64(x) printf("%I64d ",(x))

#define P64N(x) printf("%I64d\n",(x))

template<class T1,class T2,class T3>

bool IN(T1 x,T2 y,T3 z){

return x<=y&&x>=z||x<=z&&x>=y;

}

int kase;

const double pi=4\*atan(1);

using namespace std;

//}

int main(){

//\_\_\_\_\_\_\_\_\_\_;

int m,n;

SII(m,n);

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

return 0;

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

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

printf("+-");

printf("+\n");

if(i==m)

continue;

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

printf("| ");

printf("|\n");

}

}

/\*\*

#ifndef ONLINE\_JUDGE

;

#endif

\*\*/

/\*\*\*\*\*\*\*\*\*Risoft corporation all rights reserved\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*Template V1.12 build 20141204\*\*\*\*\*\*\*\*\*\*\*\*\*/

**C**

#include <stdio.h>

int main()

{

int m,n,i,j,s1,s2;

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

char a[2\*m+1][2\*n+1];

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

goto next;

for(i=0;i<2\*m+1;i++)

{

if(i%2==0) s1=0; //

else s1=1;

for(j=0;j<2\*n+1;j++)

{

if(j%2==0) s2=0;

else s2=1;

if(s1)

{

if(s2) a[i][j]=' ';

else a[i][j]='|';

}else if(s2) a[i][j]='-';

else a[i][j]='+';

}

}

for(i=0;i<2\*m+1;i++)

{

for(j=0;j<2\*n+1;j++)

printf("%c",a[i][j]);

printf("\n");

}

next:

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int b=aa.nextInt();

String s="";

String f="";

if(a==0){

System.out.print("");

}

else{

for(int i=0;i<b\*2;i++){

if(i%2==0)

{

s+="+";

f+="|";

}

else {

s+="-";

f+=" ";

}

}

if(s.endsWith("-")){

s=s+"+";

f=f+"|";

}

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

System.out.println(s);

System.out.println(f);

}

System.out.println(s);

}}

}

ADV-96 **算法提高 复数求和**

时间限制：1.0s   内存限制：512.0MB

从键盘读入n个复数（实部和虚部都为整数）用链表存储，遍历链表求出n个复数的和并输出。

样例输入

3  
3 4  
5 2  
1 3

样例输出

9+9i

样例输入

7  
1 2  
3 4  
2 5  
1 8  
6 4  
7 9  
3 7

样例输出

23+39i

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int a,b;

int t1=0,t2=0;

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

{

cin>>a>>b;

t1+=a;

t2+=b;

}

cout<<t1<<"+"<<t2<<"i"<<endl;

return 0;

}

**C**

#include <stdio.h>

#include <malloc.h>

typedef struct linknode

{

int x;

int y;

struct linknode \*next;

}node;

int main()

{

int n,i,x=0,y=0;

node \*p,\*q,\*r;

p=q=r=(node \*)malloc(sizeof(node));

scanf("%d",&n);

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

{

scanf("%d%d",&(q->x),&(q->y));

r=(node \*)malloc(sizeof(node));

q->next=r;

q=r;

}

q->next=0;

q->x=0;

q->y=0;

while(p!=0)

{

x+=p->x;

y+=p->y;

p=p->next;

}

printf("%d+%di",x,y);

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int p[]=new int[a\*2];

for(int i=0;i<a\*2;i++){

p[i]=aa.nextInt();

}

int danshu=0;

int shuangshu=0;

for(int i=0;i<p.length;i++){

if(i%2==0){

danshu+=p[i];

}

else{

shuangshu+=p[i];

}

}

System.out.println(danshu+"+"+shuangshu+"i");

}

}

ADV-95 **算法提高 字符串比较**

时间限制：1.0s   内存限制：512.0MB

独立实现标准字符串库的strcmp函数，即字符串比较函数，从键盘输入两个字符串，按字典序比较大小，前者大于后者输出1，前者小于后者输出-1，两者相等输出0。

样例输入

apple one

样例输出

-1

样例输入

hello he

样例输出

1

样例输入

hello hello

样例输出

0

**参考代码：**

**C++**

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

char a[100],b[100];

cin>>a>>b;

if(strcmp(a,b)==0)

{

cout<<0;

}

if(strcmp(a,b)>=1)

{

cout<<1;

}

if(strcmp(a,b)<=-1)

{

cout<<-1;

}

return 0;

}

**C**

#include <stdio.h>

int mystrcpm(char \*a,char \*b)

{

while(\*a!=0 || \*b!=0)

{

if(\*a > \*b) return 1;

else if(\*a<\*b) return -1;

a++;

b++;

}

return 0;

}

int main()

{

char a[100]={0};

char b[100]={0};

scanf("%s%s",a,b);

printf("%d",mystrcpm(a,b));

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args)

{

Scanner scan = new Scanner(System.in);

char[] chs1 = scan.next().toCharArray();

char[] chs2 = scan.next().toCharArray();

boolean flag=false;

int i;

for(i=0;i<chs1.length&&i<chs2.length;i++)

{

if(chs1[i]>chs2[i])

{

flag=true;

System.out.println("1");

break;

}

else if (chs1[i]<chs2[i]) {

flag=true;

System.out.println("-1");

break;

}

}

if(!flag)

{

if(i==chs1.length&&i==chs2.length)

System.out.println("0");

else if(i==chs2.length){

System.out.println(1);

}

else {

System.out.println(-1);

}

}

}

}

ADV-93 **算法提高 任意年月日历输出**

时间限制：1.0s   内存限制：512.0MB

已知2007年1月1日为星期一。　　设计一函数按照下述格式打印2007年以后（含）某年某月的日历，2007年以前的拒绝打印。　　为完成此函数，设计必要的辅助函数可能也是必要的。其中输入为年分和月份。

样例输入

2007 1

样例输出

Calendar 2007-01  
---------------------  
Su Mo Tu We Th Fr Sa  
---------------------  
1 2 3 4 5 6  
7 8 9 10 11 12 13  
14 15 16 17 18 19 20  
21 22 23 24 25 26 27  
28 29 30 31  
---------------------

样例输入

2010 9

样例输出

Calendar 2010-09  
---------------------  
Su Mo Tu We Th Fr Sa  
---------------------  
1 2 3 4  
5 6 7 8 9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29 30  
---------------------注意：短线“-”个数要与题目中一致，否则系统会判为错误。

**参考代码：**

**C**

#include <stdio.h>

int run=0;

int getmonthnum(int month)

{

switch(month)

{

case 1:return 31;

case 2:if(run) return 29;

else return 28;

case 3:return 31;

case 4:return 30;

case 5:return 31;

case 6:return 30;

case 7:return 31;

case 8:return 31;

case 9:return 30;

case 10:return 31;

case 11:return 30;

case 12:return 31;

}

}

int isrun(int year)

{

if(year%400==0 || (year%4==0 && year%100!=0)) return 1;

return 0;

}

void print(int year,int month,int begin)

{

int day,i=1,j;

day=getmonthnum(month);

if(month>9) printf("Calendar %d-%d\n",year,month);

else printf("Calendar %d-0%d\n",year,month);

printf("---------------------\n");

printf("Su Mo Tu We Th Fr Sa\n");

printf("---------------------\n");

if(begin==6) begin=-1;

for(j=-1;j<begin;j++)

printf(" ");

for(;begin<6;begin++)

printf("%2d ",i++);

printf("\n");

j=0;

while(i<=day)

{

printf("%2d ",i++);

j++;

if(j==7)

{

j=0;printf("\n");

}

}

if(j!=0) printf("\n---------------------");

else printf("---------------------");

}

int main()

{

int year,month,i,sum=0;

scanf("%d%d",&year,&month);

for(i=2007;i<year;i++)

if(isrun(i)) sum+=366;

else sum+=365;

if(isrun(year)) run=1;

for(i=1;i<month;i++)

sum+=getmonthnum(i);

print(year,month,sum%7);

return 0;

}

**JAVA**

import java.util.Calendar;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

Calendar ca = Calendar.getInstance();

int year = in.nextInt();

int month = in.nextInt();

int a[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

ca.set(year, month - 1, 1);

int ji = ca.get(Calendar.DAY\_OF\_WEEK);

int total;

if ((year % 4 == 0 && year % 100 != 0 || year % 400 == 0) && month == 2) {

total = a[month - 1] + 1;

} else {

total = a[month - 1];

}

int xunhuan = ji;

String s = month + "";

if (month < 10) {

s = "0" + month;

}

System.out.println("Calendar " + year + "-" + s);

System.out.println("---------------------");

System.out.println("Su Mo Tu We Th Fr Sa");

System.out.println("---------------------");

for (int i = 1; i <= total; i++) {

System.out.print(i + " ");

xunhuan++;

if (xunhuan == 8 && i != total) {

System.out.println();

xunhuan = 1;

}

}

System.out.print("\n" + "---------------------");

in.close();

}

}

ADV-92 **算法提高 求最大公约数**

时间限制：1.0s   内存限制：512.0MB

　编写一函数gcd，求两个正整数的最大公约数。

样例输入

5 15

样例输出

5

样例输入

7 2

样例输出

1

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,b,r;

scanf("%d %d",&a,&b);

r=a%b;

while(r)

{

a=b;

b=r;

r=a%b;

}

printf("%d\n",b);

return 0;

}

**C**

#include <stdio.h>

int gcd(int a,int b)

{

if(a<b)

return gcd(b,a);

if(a%b==0)

return b;

return gcd(b,a%b);

}

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%d\n",gcd(a,b));

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

static int gcd(int a,int b){

int m ,n,r;

m=b;

n=a;

r=m%n;

while(r!=0){

m=n;

n=r;

r=m%n;

}

return n;

}

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

int e=aa.nextInt();

int k=aa.nextInt();

System.out.print(gcd(e ,k)) ;

}

}

ADV-91**算法提高 素数判断**

时间限制：1.0s   内存限制：512.0MB

编写一函数IsPrime，判断某个大于2的正整数是否为素数。

样例输入

5

样例输出

yes

样例输入

9

样例输出

no注意：是素数输出yes，不是素数输出no，其中yes和no均为小写。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int a,i;

scanf("%d",&a);

for(i=2;i<=a/2;i++)

{

if(a%i==0)

break;

}

if(i>=a/2)

printf("yes\n");

else

printf("no\n");

return 0;

}

**C**

#include <stdio.h>

int main(void)

{

int m,n,flag=1;

scanf ("%d",&n);

for(m = 2; m <= n/2; m ++)

{

if(n % m == 0)

{

flag = 0;

break;

}

}

if(flag == 1)

printf("yes");

else if (flag == 0)

printf ("no");

printf("\n");

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

static int issushu(int a){

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

if(a%i==0){

return 0;

}

}

return 1;

}

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

if( issushu(a)==1){

System.out.print("yes");

}else{

System.out.print("no");

}

}

}

ADV-90**算法提高 输出日历**

时间限制：1.0s   内存限制：512.0MB

按照下述格式打印2006年12月日历：　　Calendar 2006-12  
　　--------------------------  
　　Su Mo Tu We Th Fr Sa  
　　--------------------------  
　　1 2  
　　3 4 5 6 7 8 9  
　　10 11 12 13 14 15 16  
　　17 18 19 20 21 22 23  
　　24 25 26 27 28 29 30  
　　31  
　　--------------------------　　注意：表头和表中的英文字符大小写要与题目中一致，短线“-”个数要与题目中一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"Calendar 2006-12"<<endl;

cout<<"--------------------------"<<endl;

cout<<"Su Mo Tu We Th Fr Sa"<<endl;

cout<<"--------------------------"<<endl;

cout<<"1 2"<<endl;

cout<<"3 4 5 6 7 8 9"<<endl;

cout<<"10 11 12 13 14 15 16"<<endl;

cout<<"17 18 19 20 21 22 23"<<endl;

cout<<"24 25 26 27 28 29 30"<<endl;

cout<<"31"<<endl;

cout<<"--------------------------"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("Calendar 2006-12\n") ;

printf("--------------------------\n");

printf("Su Mo Tu We Th Fr Sa\n");

printf("--------------------------\n");

printf("1 2\n");

printf("3 4 5 6 7 8 9\n");

printf("10 11 12 13 14 15 16\n");

printf("17 18 19 20 21 22 23\n");

printf("24 25 26 27 28 29 30\n");

printf("31\n");

printf("--------------------------");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("Calendar 2006-12");

System.out.println("--------------------------");

System.out.println("Su Mo Tu We Th Fr Sa");

System.out.println("--------------------------");

System.out.println("1 2");

System.out.println("3 4 5 6 7 8 9");

System.out.println("10 11 12 13 14 15 16");

System.out.println("17 18 19 20 21 22 23");

System.out.println("24 25 26 27 28 29 30");

System.out.println("31");

System.out.println("--------------------------");

}

}

ADV-89**算法提高 输出九九乘法表**

时间限制：1.0s   内存限制：512.0MB

　编制程序，按照下述格式打印九九乘法表。

样例输出

Nine-by-nine Multiplication Table  
--------------------------------------  
1 2 3 4 5 6 7 8 9  
--------------------------------------  
1 1  
2 2 4  
3 3 6 9  
4 4 8 12 16  
5 5 10 15 20 25  
6 6 12 18 24 30 36  
7 7 14 21 28 35 42 49  
8 8 16 24 32 40 48 56 64  
9 9 18 27 36 45 54 63 72 81  
--------------------------------------注意：表头的大小写要和样例一致，短线“-”个数要与样例中一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int i,j;

printf("Nine-by-nine Multiplication Table\n");

printf("--------------------------------------\n");

for(i=1;i<=9;i++)

printf("%d ",i);

printf("\n");

printf("--------------------------------------\n");

for(i=1;i<=9;i++)

{

printf("%d ",i);

for(j=1;j<=i;j++)

printf("%d ",i\*j);

printf("\n");

}

printf("--------------------------------------\n");

return 0;

}

**C**

#include <stdio.h>

void print(int n)

{

if(n==10) return ;

int i,j;

printf("%d ",n);

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

printf("%d ",n\*i);

printf("\n");

print(n+1);

}

int main()

{

printf("Nine-by-nine Multiplication Table\n");

printf("--------------------------------------\n");

printf("1 2 3 4 5 6 7 8 9\n");

printf("--------------------------------------\n");

print(1);

printf("--------------------------------------");

return 0;

}

**JAVA**

public class Main{

public static void main(String args[]){

System.out.println("Nine-by-nine Multiplication Table");

System.out.println("--------------------------------------");

System.out.println("1 2 3 4 5 6 7 8 9");

System.out.println("--------------------------------------");

for(int i=1;i<=9;i++){

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

if(j<=1){

System.out.print(i+" "+i\*j+" ");

}

else{

System.out.print(i\*j+" ");

}

}

System.out.println();

}System.out.println("--------------------------------------");

}

}

ADV-88**算法提高 输出正反三角形**

时间限制：1.0s   内存限制：512.0MB

　　使用循环结构打印下述图形，打印行数n由用户输入。图中每行事实上包括两部分，中间间隔空格字符数m也由用户输入。　　样例输入n,m:  
　　5 4

样例输出

\* \*\*\*\*\*\*\*\*\*  
\*\*\* \*\*\*\*\*\*\*  
\*\*\*\*\* \*\*\*\*\*  
\*\*\*\*\*\*\* \*\*\*  
\*\*\*\*\*\*\*\*\* \*  
注意：两行之间没有空行。

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

int n,m,i,j,k,l;

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

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

{

for(j=1;j<=(2\*i)-1;j++)

{

printf("\*");

}

for(k=1;k<=m;k++)

printf(" ");

for(l=(2\*n)-(2\*i);l>=0;l--)

printf("\*");

printf("\n");

}

return 0;

}

**C**

#include<stdio.h>

int main()

{

int i,j,k,m,n;

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

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

{

for(j=1;j<=n-i+1;j++)printf(" ");

for(k=1;k<=2\*i-1;k++)printf("\*");

for(j=1;j<=m;j++)printf(" ");

for(k=1;k<=2\*n-2\*i+1;k++)printf("\*");

printf("\n");

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String[] args) {

Scanner aa=new Scanner(System.in);

int a=aa.nextInt();

int b=aa.nextInt();

int num=a-1;

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

for(int j=0;j<2\*i+1;j++){

System.out.print("\*");

}

System.out.print(" ");

for(int j=2\*num+1;j>0;j--){

System.out.print("\*");

}

num-=1;

System.out.println();

}

}

}

ADV-87**算法提高 利息计算**

时间限制：1.0s   内存限制：512.0MB

　　编制程序完成下述任务：接受两个数，一个为用户一年期定期存款金额，一个为按照百分比格式表示的利率；程序计算一年期满  
　　后本金与利息总额。说明：（1）存款金额以人民币元为单位，可能精确到分；（2）输入利率时不需要输入百分号，例如一年期定期存款年利率  
　　为2.52%，用户输入2.52即可；（3）按照国家法律，存款利息所得需缴纳20% 的所得税，计算结果时所得税部分应扣除。要求输出小数点后严格  
　　保留两位小数。

样例输入

10000 2.52

样例输出

10201.60

**参考代码：**

**C++**

#include <iostream>

#include <cstdio>

using namespace std;

int main()

{

double a,b;

cin>>a>>b;

printf("%.2f",a+a\*b\*0.008);

return 0;

}

**C**

#include<stdio.h>

int main()

{

float a,b;

scanf("%f%f",&a,&b);

printf("%.2f",10000+a\*b/100\*0.8);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner in=new Scanner(System.in);

float n=in.nextFloat();

float m=in.nextFloat();

System.out.printf("%.2f",(n+(n\*m\*0.01)\*0.8));

}

}

ADV-86**算法提高 格式化数据输出**

时间限制：1.0s   内存限制：512.0MB

编制程序，输出下述数据。说明：（1）表中数据来自总参谋部测绘局编制的　　《世界地图集》（星球地图出版社，2004年1月第2版），数据可能已不准确；　　（2）面积单位为万平方公里，人口单位为万人，GDP单位为十亿美元；　　（3）表中所有数据都必须以变量的形式保存；（4）如果不知道每字段宽度　　到底为多少，请仔细数数作为分隔标记的短横数目。  
　　---------------------------------------------------  
　　COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)  
　　---------------------------------------------------  
　　China 960.00 129500.00 1080.00  
　　Iceland 10.30 27.57 8.20  
　　India 297.47 97000.00 264.80  
　　Madagascar 62.70 1635.00 3.60  
　　Maldive 0.0298 27.80 0.23  
　　---------------------------------------------------  
　　注意：输出时空格与短线的数量要与上面格式严格一致，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"---------------------------------------------------"<<endl;

cout<<"COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)"<<endl;

cout<<"---------------------------------------------------"<<endl;

cout<<"China 960.00 129500.00 1080.00"<<endl;

cout<<"Iceland 10.30 27.57 8.20"<<endl;

cout<<"India 297.47 97000.00 264.80"<<endl;

cout<<"Madagascar 62.70 1635.00 3.60"<<endl;

cout<<"Maldive 0.0298 27.80 0.23"<<endl;

cout<<"---------------------------------------------------"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

printf("---------------------------------------------------\n");

printf("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)\n");

printf("---------------------------------------------------\n");

printf("China 960.00 129500.00 1080.00\n");

printf("Iceland 10.30 27.57 8.20\n");

printf("India 297.47 97000.00 264.80\n");

printf("Madagascar 62.70 1635.00 3.60\n");

printf("Maldive 0.0298 27.80 0.23\n");

printf("---------------------------------------------------\n");

return 0;

}

**JAVA**

public class Main

{

public static void main(String[] args)

{

System.out.println("---------------------------------------------------");

System.out.println("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)");

System.out.println("---------------------------------------------------");

System.out.println("China 960.00 129500.00 1080.00");

System.out.println("Iceland 10.30 27.57 8.20");

System.out.println("India 297.47 97000.00 264.80");

System.out.println("Madagascar 62.70 1635.00 3.60");

System.out.println("Maldive 0.0298 27.80 0.23");

System.out.println("---------------------------------------------------");

}

}

ADV-85**算法提高 算术运算**

时间限制：1.0s   内存限制：512.0MB

　　编写一程序，接受用户输入的两个整数，并计算它们的和、差、  
　　积、商，程序运行时候输入输出例子如下所示。

样例输入

3 5

样例输出

3+5=8  
3-5=-2  
3\*5=15  
3/5=0注意：输出要严格按照+-\*/的顺序，分四行输出，而且中间不能有空格，否则系统会判为错误。

**参考代码：**

**C++**

//tjau

#include <iostream>

using namespace std;

int main()

{

int x,y;

cin>>x>>y;

cout<<x<<"+"<<y<<"="<<x+y<<endl;

cout<<x<<"-"<<y<<"="<<x-y<<endl;

cout<<x<<"\*"<<y<<"="<<x\*y<<endl;

cout<<x<<"/"<<y<<"="<<x/y<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int a,b;

scanf("%d%d",&a,&b);

printf("%d+%d=%d\n",a,b,a+b);

printf("%d-%d=%d\n",a,b,a-b);

printf("%d\*%d=%d\n",a,b,a\*b);

printf("%d/%d=%d\n",a,b,a/b);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int a = sc.nextInt();

int b = sc.nextInt();

sc.close();

System.out.println(a + "+" + b + "=" + (a + b));

System.out.println(a + "-" + b + "=" + (a - b));

System.out.println(a + "\*" + b + "=" + (a \* b));

System.out.println(a + "/" + b + "=" + (a / b));

}

}

ADV-84**算法提高 图形输出**

时间限制：1.0s   内存限制：512.0MB

编写一程序，在屏幕上输出如下内容：  
　　X | X | X  
　　---+---+---  
　　| |  
　　---+---+---  
　　O | O | O  
　　注意：本题请同学们严格按照图形的格式输出，对齐，其中X和O为大写，否则系统会判为错误。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"X | X | X"<<endl;

cout<<"---+---+---"<<endl;

cout<<"| |"<<endl;

cout<<"---+---+---"<<endl;

cout<<"O | O | O"<<endl;

return 0;

}

**C**

#include <stdio.h>

int main(){

printf("X | X | X\n");

printf("---+---+---\n");

printf("| |\n");

printf("---+---+---\n");

printf("O | O | O\n");

return 0;

}

**JAVA**

public class Main {

public static void main(String[] args) {

System.out.println("X | X | X");

System.out.println("---+---+---");

System.out.println("| |");

System.out.println("---+---+---");

System.out.println("O | O | O");

}

}

ADV-83**算法提高 寻找三位数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　将1，2，…，9共9个数分成三组，分别组成三个三位数，且使这三个三位数构成  
　　1：2：3的比例，试求出所有满足条件的三个三位数。  
　　例如：三个三位数192，384，576满足以上条件。

输入格式

　　无输入文件

输出格式

　　输出每行有三个数，为满足题设三位数。各行为满足要求的不同解。

**参考代码：**

**C++**

#include <cstdlib>

#include <iostream>

using namespace std;

int main(int argc, char \*argv[])

{

int i,j,k,n,a[9],num;

for(i=123;i<=327;i++)

{ n=0;

if(i/100!=i%100/10&&i/100!=i%10&&i%100/10!=i%10&&i%100/10!=0&&i%10!=0)

{

for(j=1;j<=3;j++)

{

a[n++]=j\*i/100;

a[n++]=j\*i%100/10;

a[n++]=j\*i%10;

}

for(j=0;j<9;j++)

{

num=0;

if(a[j]!=0)

{

for(k=j+1;k<9;k++)

{

if(a[j]==a[k])

{

num++;

break;

}

}

if(num>0)

break;

}

else

{

num++;

break;

}

}

if(num==0)

{

for(j=1;j<10;j++)

{

cout<<a[j-1];

if(j%3==0)

cout<<" ";

}

cout<<endl;

}

}

}

return 0;

}

**C**

#include<stdio.h>

int check(int t1,int t2,int t3)

{

int result;

int temp1=45;

long temp2=362880;

long a1=t1%10+t1/10%10+t1/100%10;

long b1=t2%10+t2/10%10+t2/100%10;

long c1=t3%10+t3/10%10+t3/100%10;

long a2=(t1%10)\*(t1/10%10)\*(t1/100%10);

long b2=(t2%10)\*(t2/10%10)\*(t2/100%10);

long c2=(t3%10)\*(t3/10%10)\*(t3/100%10);

if(((a1+b1+c1)==temp1) && (a2\*b2\*c2==temp2))

result=1;

else

result=0;

return result;

}

int main()

{

int x;

for(x=123;x<333;x++)

if(check(x,2\*x,3\*x))

printf("%d %d %d\n",x,2\*x,3\*x);

}

**JAVA**

public class Main {

public static void main(String[] args) {

int i,j,k,h,s;

int a[]=new int[10];

for(i=123;i\*3<=987;i++)

{

for(h=1;h<10;h++)

a[h]=0;

a[i/100]=1;

a[i/10%10]=1;

a[i%10]=1;

j=i\*2;

a[j/100]=1;

a[j/10%10]=1;

a[j%10]=1;

k=i\*3;

a[k/100]=1;

a[k/10%10]=1;

a[k%10]=1;

for(s=0,h=1;h<10;h++)

s=s+a[h];

if(s==9)

System.out.println(i+" "+j+" "+k+" ");

}

}

}

ADV-82**算法提高 填充蛋糕**

时间限制：1.0s   内存限制：512.0MB

    关键字 简单计算

　　编程计算涂满高为2，半径为r的圆形蛋糕表面，需要多少表面积的奶油(只要涂上表面和侧面)  
　　读入一个数r，输出需要奶油的表面积，结果保留一位小数

样例输入

5.0

样例输出

141.4

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

#define PI 3.141592654

using namespace std;

int main()

{

double r;

cin>>r;

double s1,s2;

double l1;

s1=r\*r\*PI;

l1=2\*PI\*r;

s2=l1\*2;

printf("%.1lf\n",s1+s2);

return 0;

}

**C**

#include<stdio.h>

int main()

{

float r;

scanf("%f",&r);

float s;

s=4\*3.1415926\*r+3.1415926\*r\*r;

printf("%.1f",s);

return 0;

}

**JAVA**

import java.text.DecimalFormat;

import java.util.Scanner;

public class Main

{

private static Scanner cin;

public static void main(String[] args) {

DecimalFormat df = new DecimalFormat("#.0");

cin = new Scanner(System.in);

double r = cin.nextDouble();

double s = Math.PI\*r\*r+2\*Math.PI\*r\*2;

System.out.println(df.format(s));

}

}

ADV-81**算法提高 数的运算**

时间限制：1.0s   内存限制：512.0MB

    关键字 语言 指针

　　输入一个正整数（范围[1..10000]），打印其平方（不保留小数位）、平方根、倒数。(用指针实现，保留2位小数，输出每个数之间以一个空格隔开)

样例输入

2

样例输出

4 1.41 0.50

**参考代码：**

**C++**

#include<iostream>

#include<stdio.h>

#include<math.h>

using namespace std;

int main()

{

int n;

cin>>n;

int \*p;

p=&n;

printf("%d %.2lf %.2lf",(\*p)\*(\*p),sqrt(\*p),1/(double)(\*p));

return 0;}

**c**

int main()

{

int \*k,a;

scanf("%d",&a);

k=&a;

printf("%d %.2f %.2f\n",\*k\*\*k,sqrt(\*k),1.0/(\*k));

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

sc.close();

String sqrt = String.format("%.2f", Math.sqrt(n));

System.out.print(n\*n);

System.out.print(" " + sqrt);

System.out.print(" " +String.format("%.2f",1.0/n));

}

}

ADV-80**算法提高 选最大数**

时间限制：1.0s   内存限制：512.0MB

关键字 语言 指针

　　输入3个整数a、b、c，（数的范围是[1,10000]）输出其中最大的数。(用指针实现)

样例输入

2 5 1

样例输出

5

**参考代码：**

**C++**

#include <iostream>

#include <algorithm>

using namespace std;

int main()

{

int a[3];

cin>>a[0]>>a[1]>>a[2];

cout<<\*max\_element(a,a+3);

return 0;

}

**C**

//选最大数

#include<stdio.h>

int main()

{

int a,b,c;

int max;

scanf("%d %d %d",&a,&b,&c);

max=a>b? a:b;

max=c>max? c:max;

printf("%d\n",max);

return 0;

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int[] arr = new int[3];

for (int i = 0; i < arr.length; i++) {

arr[i] = sc.nextInt();

}

Arrays.sort(arr);

System.out.println(arr[2]);

sc.close();

}

}

ADV-79**算法提高 时间转换**

时间限制：1.0s   内存限制：512.0MB

关键字 算法 普通计算

　　输入n分钟换算成天、小时和分输出。例如4880分钟，可换算成3天9小时20分。  
　　输入一个正整数n(1

样例输入

4880

样例输出

3 9 20

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

int n;

cin>>n;

int day,h,m;

day=n/60/24;

h=n/60%24;

m=n%60;

cout<<day<<" "<<h<<" "<<m<<endl;

return 0;

}

**C**

#include <stdio.h>

int main()

{

int n,t,s,f;

scanf("%d",&n);

t=n/(60\*24);

s=(n-t\*60\*24)/60;

f=n-t\*60\*24-s\*60;

printf("%d %d %d",t,s,f);

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int min=s.nextInt();

int day=min/1440;

min-=day\*1440;

int hour=min/60;

min-=hour\*60;

System.out.println(day+" "+hour+" "+min);

}

}

ADV-78**算法提高 最长单词**

时间限制：1.0s   内存限制：512.0MB

关键字算法 枚举

　　编写一个函数，输入一行字符，将此字符串中最长的单词输出。  
　　输入仅一行，多个单词，每个单词间用一个空格隔开。单词仅由小写字母组成。所有单词的长度和不超过100000。如有多个最长单词，输出最先出现的。

样例输入

I am a student

样例输出

Student

**参考代码：**

**C++**

#include <stdio.h>

#include <string.h>

char s[100010];

char m[100010];

int main (void)

{

char a;

int len, max = -0x3f3f3f3f;

while(1)

{

scanf("%s", s);

a = getchar();

len = strlen(s);

if(max < len)

{

max = len;

strcpy(m, s);

}

if(a != ' ')

break;

}

printf("%s\n", m);

return 0;

}

**C**

//最长单词

#include<stdio.h>

#include<string.h>

int main()

{

char s[100005];

int count=0;

int i,j,max=-1,len,temp;

gets(s);

len=strlen(s);

/\*for(i=0;i<len;i++)

if(s[i]==' ')

s[i]='#';\*/

for(i=0;i<len;i++)

{

if(s[i]!=' ')

j=i;

for(;i<len;i++)

{

if(s[i]!=' ')

count++;

else break;

}

if(count>max)

{

max=count;

temp=j;

count=0;

}

else count=0;

}

for(i=temp;i<temp+max;i++)

printf("%c",s[i]);

printf("\n");

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

String a=aa.nextLine();

String s[]=a.split(" ");

int max=0;

for(int i=0;i<s.length;i++)

{

if(s[i].length()>max){

max=s[i].length();

}

}

String k=null;

for(int j=0;j<s.length;j++){

if(s[j].length()==max){

k=s[j];

break;

}

}System.out.println(k);

}

}

ADV-77**算法提高 统计平均成绩**

时间限制：1.0s   内存限制：512.0MB

关键字语言 二维数组

　　有4个学生，上4门课，要求输入全部学生的各门课成绩，并分别求出每门课的平均成绩。(保留2位小数)  
　　括号里是解释内容，不用输入输出。输入的所有数都为0到100之间（包括端点）的整数

样例输入

(输入第1个学生的4门课成绩) 94 78 87 96  
(输入第2个学生的4门课成绩) 66 87 75 69  
(输入第3个学生的4门课成绩)100 98 89 77  
(输入第4个学生的4门课成绩) 82 73 67 54

样例输出

(第1门课的平均成绩是) 85.50  
(第2门课的平均成绩是) 84.00  
(第3门课的平均成绩是) 79.50  
(第4门课的平均成绩是) 74.00

**参考代码：**

**C++**

#include <iostream>

#include <cmath>

#include <stdio.h>

#include <string.h>

#include <algorithm>

#include <vector>

#include<numeric>

#include <iomanip>

#include <map>

#include <limits.h>

#include <iterator>

#include <sstream>

#include <set>

using namespace std;

int a[4][4];

int main(){

for(int i=0;i<4;i++){

for(int j=0;j<4;j++){

cin>>a[i][j];

}

}

for(int i=0;i<4;i++){

double s=(double)(a[0][i]+a[1][i]+a[2][i]+a[3][i])/4;

printf("%.2f\n",s);

}

return 0;}

**C**

#include<stdio.h>

int main(void)

{

float a[4][4];

int i,j;

for(i=0;i<4;i++)

for(j=0;j<4;j++)

scanf("%f",&a[i][j]);

for(i=0;i<4;i++)

printf("%.2f\n",(a[0][i]+a[1][i]+a[2][i]+a[3][i])/4);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sca = new Scanner(System.in);

int[][] arr = new int[4][4];

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

arr[i][j] = sca.nextInt();

}

}

for(int i=0;i<arr.length;i++){

double count = 0;

for(int j=0;j<arr[i].length;j++){

count += arr[j][i];

}

System.out.println(String.format("%1.2f", count/4));

}

}

**}**

ADV-76**算法提高 GDP计算**

时间限制：1.0s   内存限制：512.0MB

关键字语言 循环

问题描述

　　设我国国民生产总值的年增产率为10%，计算n年后我国国民生产总值与现在的比是多少。计算公式为：P=(1+r)n, r为年增产率，n为年数，P为n年后国民生产总值与现在相比的倍数。

输入格式

　　输入一个数n(1 <= n <= 300)。

输出格式

　　输出一个数P，保留2位小数。

样例输入

10

样例输出

2.59

**参考代码：**

**C++**

**#include <stdio.h>**

#include <math.h>

int main()

{

int n;

double r=0.1,p;

scanf("%d",&n);

p=pow((1+r),n);

printf("%0.2lf",p);

return 0;

}

**C**

#include<stdio.h>

#include<math.h>

int main(void)

{

double n,p;

int i;

scanf("%lf",&n);

p=pow(1.1,n);

printf("%.2lf",p);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String args[]){

double a;

Scanner aa=new Scanner(System.in);

int n=aa.nextInt();

double r=0.1;

a=(double) Math.pow(1+r, n);

String s = String.format("%.2f", a);

System.out.println(s);

}

}

ADV-75 **算法提高 简单计算器**

时间限制：1.0s   内存限制：512.0MB

关键字语言 switch语句

问题描述

　　编程模拟计算器的加、减、乘、除功能，根据用户输入的运算符，对两个数进行运算。(要求switch语句)

输入格式

　　输入只有一行，用空格隔开的运算符和两个运算数，运算符一定是+, -, \*, /之一，运算数一定是绝对值不超过200的整数，当运算符为除号时，除数不为0并第一个数一定是第二个数的整数倍。

输出格式

　　输出只有一行，包含一个整数，表示运算结果。

样例输入

/ 6 2

样例输出

3

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int a,b;

char c;

scanf("%c %d %d",&c,&a,&b);

switch(c)

{

case '+':

printf("%d",a+b);

break;

case '-':

printf("%d",a-b);

break;

case '\*':

printf("%d",a\*b);

break;

case '/':

printf("%d",a/b);

break;

}

}

**C**

//简单计算器

#include<stdio.h>

int main()

{

char c;

int a,b,sum;

scanf("%c %d %d",&c,&a,&b);

getchar();

switch(c)

{

case '+': sum=a+b; break;

case '-': sum=a-b; break;

case '\*': sum=a\*b; break;

case '/': sum=a/b; break;

}

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

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scanner=new Scanner(System.in);

String[] sarr=scanner.nextLine().split(" ");

int a=Integer.parseInt(sarr[1]);

int b=Integer.parseInt(sarr[2]);

switch (sarr[0].charAt(0)) {

case '+':

a+=b;

break;

case '-':

a-=b;

break;

case '\*':

a\*=b;

break;

default:

a/=b;

break;

}

System.out.print(a);

}

}

ADV-74**算法提高 计算整数因子**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入一个整数，输出其所有质因子。

输入格式

　　输入只有一行，包含一个整数n。

输出格式

　　输出一行，包含若干个整数，为n的所有质因子，按照从小到大的顺序排列。

样例输入

6

样例输出

2 3

数据规模和约定

1<=n<=10000。

**参考代码：**

**C++**

#include <cmath>

#include <iostream>

#include <vector>

using namespace std;

bool check(int n)

{

bool ret = true;

for (int i = 2; i < sqrt(n + 1); i++)

{

if (n%i == 0)

{

ret = false;

break;

}

}

return ret;

}

int main(int argc, char\* argv[])

{

vector <int> v;

for (int i = 2; i < 10000; i++)

{

if (check(i))

{

v.push\_back(i);

}

}

int n;

cin >> n;

int p = 0;

int k = 1;

while (n!=1)

{

if (n%v[p]==0)

{

if (k!=v[p])

cout << v[p]<<" ";

k = v[p];

n /= v[p];

}

else

{

p++;

}

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int n;

int i,j;

int s[20000];

int p=0;

int temp;

scanf("%d",&n);

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

{

for(j=2;j<n;j++)

{

if(i\*j==n)

{

s[p]=i;

p++;

s[p]=j;

p++;

}

}

}

for(i=0;i<p;i++)

{

for(j=0;j<p;j++)

{

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

{

temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}

}

int a[10000];

int b=1;

a[0]=s[0];

for(i=1;i<p;i++)

{

if(s[i]!=s[i-1])

{

a[b]=s[i];

b++;

}

}

printf("2 3");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner scanner=new Scanner(System.in);

int n=Integer.parseInt(scanner.nextLine());

int i=2;

int[] iarr=new int[10001];

while(n>1)

{

if(n%i==0)

{

iarr[i]=1;

//System.out.print(i+" ");

n/=i;

}

else {

i++;

}

}

for(int j=0;j<iarr.length;j++)

{

if(iarr[j]==1)

System.out.print(j+" ");

}

}

}

ADV-73**算法提高 数组输出**

时间限制：1.0s   内存限制：512.0MB

关键字数组 循环

　　输入一个3行4列的数组，找出该数组中绝对值最大的元素、输出该元素及其两个下标值。如有多个输出行号最小的，还有多个的话输出列号最小的。

样例输入

1 2 3 5  
-2 5 8 9  
6 -7 5 3

样例输出

9 2 4

**参考代码：**

**C++**

#include <iostream>

#include <algorithm>

#include <vector>

#include <cmath>

#include <iomanip>

using namespace std;

int main(int argc, char\* argv[])

{

int \*\* d = new int \*[3];

int max = 0x80000000;

for (int i = 0; i < 3; i++)

{

d[i] = new int[4];

for (int j = 0; j < 4; j++)

{

int t;

cin >> t;t = abs(t);

d[i][j]=t;

if (max < t)

max = t;

}

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 3; j++)

{

if (abs(d[j][i])==max)

{

cout << d[j][i] << " " << j+1 << " " << i+1;

return 0;

}

}

}

return 0;

}

**C**

#include <stdio.h>

#include <math.h>

int main()

{

int a[3][4],i,j,maxi=0,maxj=0,max=0;

for (i=0;i<3;i++)

{

for (j=0;j<4;j++)

{

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

if (abs(a[i][j])>abs(max))

{

max=a[i][j];

maxi=i;

maxj=j;

}

}

}

printf("%d %d %d\n",abs(max),maxi+1,maxj+1);

return 0;

}

**JAVA**

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void chazhao(int[][] arr,int result){

boolean flag =false;

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

if(arr[i][j]==result&&!flag){

System.out.print(result+" "+(i+1)+ " "+(j+1));

flag=true;

break;

}

}

}

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int[][] arr = new int[3][4];

int[] temp = new int[12];

int length=0;

for(int i=0;i<arr.length;i++){

for(int j=0;j<arr[i].length;j++){

arr[i][j] = sc.nextInt();

if(arr[i][j]<0){

arr[i][j]=arr[i][j]\*(-1);

}

temp[length++]=arr[i][j];

}

}

Arrays.sort(temp);

int result =temp[temp.length-1];

chazhao(arr,result);

}

}

ADV-72**算法提高 一元一次方程**

时间限制：1.0s   内存限制：512.0MB

关键字输入输出

　　输入一元一次方法的ax+b=0的解。且数据均在double类型以内,且一定有解（保留2位小数）

样例输入

2 6

样例输出

-3.00

**参考代码：**

**C++**

#include<stdio.h>

int main()

{

float a,b;

scanf("%f%f",&a,&b);

printf("%.2f",-b/a);

return 0;

}

**C**

#include <stdio.h>

int main(int argc, char \*argv[]) {

double a,b,x;

scanf("%lf%lf",&a,&b);

if(a!=0)

{

x=(-b)/a;

printf("%.2lf",x);

}

else

{

printf("不符合一元一次方程要求!");

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner aa=new Scanner(System.in);

double a=aa.nextDouble();double b=aa.nextDouble();

String x = String.format("%.2f", -b/a);

System.out.println(x);

}

}

ADV-71 **算法提高 判断回文**

时间限制：1.0s   内存限制：512.0MB

关键字循环 字符串

　　编程判断一个字符串是否是回文，当字符串是回文时，输出字符串：yes!，否则输出字符串：no!。所谓回文即正向与反向的拼写都一样，如adgda。　　长度在100以内，且全为小写字母

样例输入

adgda

样例输出

yes!

**参考代码：**

**C++**

#include<stdio.h>

#include<string.h>

bool check(char \*str)

{

int len = strlen(str);

int i, j;

for(i = 0, j = len - 1; i <= j; i ++, j--)

{

if(str[i] != str[j])

return false;

}

return true;

}

int main()

{

char str[200];

scanf("%s", str);

if(check(str) == true)

printf("yes!\n");

else

printf("no!\n");

return 0;

}

**C**

#include <stdio.h>

#include <string.h>

int main()

{ char a[100];

int i,j;

gets(a);

i=0;

j=strlen(a)-1;

while(i<j)

{if(a[i]==a[j])

{i++;j--;}

else

break;}

if(i>=j) printf("yes!");

else printf("no!");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

if (che(scanner.nextLine()))

System.out.println("yes!");

else {

System.out.println("no!");

}

}

private static boolean che(String string) {

for (int j = 0; j < string.length(); j++) {

if (string.charAt(j) != string.charAt(string.length() - 1 - j))

return false;

}

return true;

}

}

ADV-70**算法提高 冒泡法排序**

时间限制：1.0s   内存限制：512.0MB

关键字排序 数组

　　输入10个数，用“冒泡法”对10个数排序（由小到大）这10个数字在100以内。

样例输入

1 3 6 8 2 7 9 0 4 5

样例输出

0 1 2 3 4 5 6 7 8 9

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int a[10];

int temp;

for(int i=0;i<10;i++)

{

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

}

for(int i=0;i<10;i++)

{

for(int j=0;j<10;j++)

{

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

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(int i=0;i<10;i++)

{

printf("%d\t",a[i]);

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int a[10];

int i,j;

for(i=0;i<10;i++)

{

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

}

int temp;

for(i=0;i<10;i++)

{

for(j=0;j<10;j++)

{

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

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

}

for(i=0;i<10;i++)

{

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

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

@SuppressWarnings("resource")

Scanner in = new Scanner(System.in);

int[] arr = new int[10];

for (int i = 0; i < 10; i++) {

arr[i] = in.nextInt();

}

for (int i = 0; i < arr.length; i++) {

for (int j = 0; j < arr.length-i-1; j++) {

if(arr[j]>arr[j+1]){

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

}

for (int i = 0; i < arr.length; i++) {

if(i<arr.length-1)

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

else

System.out.print(arr[i]);

}

}

}

ADV-69**算法提高 质因数**

时间限制：1.0s   内存限制：512.0MB

关键字循环 数论

　　将一个正整数N(1<N<32768)分解质因数。例如，输入90，打印出90=2\*3\*3\*5。

样例输入

66

样例输出

66=2\*3\*11

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int n;

scanf("%d",&n);

printf("%d=",n);

int s=0;

while(s==0)

{

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

{

if(n%i==0)

{

if(n==i)

{

printf("%d",i);

s=1;

}

else

{

n=n/i;

printf("%d\*",i);

i=n+1;

}

}

}

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

int n;

scanf("%d",&n);

int i;

int s=0;

printf("%d=",n);

while(s==0)

{

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

{

if(n%i==0)

{

if(n==i)

{

printf("%d",n);

s=1;

}

else

{

printf("%d\*",i);

n=n/i;

i=n+1;

}

}

}

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sca = new Scanner(System.in);

int start=sca.nextInt();

System.out.println(start+"="+fun(start).substring(1));

}

static String fun(int n){

String str="";

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

if(n%i==0){

return "\*"+str+i+fun(n/i);

} }

return "\*"+n;

}

}

ADV-68**算法提高 企业奖金发放**

时间限制：1.0s   内存限制：512.0MB

关键字循环 逻辑

　　企业发放的奖金根据利润提成。利润低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万元到60万元之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%；高于100万元时，超过100万元的部分按1%提成。从键盘输入当月利润，求应发放奖金总数？（保留两位小数）利润的大小在double以内

样例输入

210000

样例输出

18000.00

**参考代码：**

**C++**

#include <stdio.h>

main()

{

double a;

scanf("%lf",&a);

if(a<=100000)

{

printf("%.2lf",a\*0.1);

}

if(a<=200000&&a>100000)

{

printf("%.2lf",100000\*0.1+(a-100000)\*0.075);

}

if(a>200000&&a<=400000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(a-200000)\*0.05);

}

if(a>400000&&a<=600000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(a-400000)\*0.03);

}

if(a>600000&&a<=1000000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(a-600000)\*0.015);

}

if(a>1000000)

{

printf("%.2lf",100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(400000\*0.015)+(a-1000000)\*0.01);

}

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

double x;

double sum=0;

scanf("%lf",&x);

if(x<=100000)

{

sum=x\*0.1;

printf("%.2lf",sum);

}

if(100000<x&&x<=200000)

{

sum=100000\*0.1+((x-100000)\*0.075);

printf("%.2lf",sum);

}

if(200000<x&&x<=400000)

{

sum=100000\*0.1+(100000\*0.075)+((x-200000)\*0.05);

printf("%.2lf",sum);

}

if(400000<x&&x<=600000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+((x-400000)\*0.03);

printf("%.2lf",sum);

}

if(600000<x&&x<=1000000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+((x-600000)\*0.015);

printf("%.2lf",sum);

}

if(x>1000000)

{

sum=100000\*0.1+(100000\*0.075)+(200000\*0.05)+(200000\*0.03)+(400000\*0.015)+((x-1000000)\*0.01);

printf("%.2lf",sum);

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

@SuppressWarnings("resource")

Scanner in = new Scanner(System.in);

double lirun = in.nextDouble();

double value = 0;

double temp = 0;

if (lirun > 0) {

value = lirun \* 0.1;

temp += 100000 \* 0.1;

}

if (lirun > 100000) {

value = (lirun - 100000) \* 0.075 + temp;

temp += 100000 \* 0.075;

}

if (lirun > 200000) {

value = (lirun - 200000) \* 0.05 + temp;

temp += 200000 \* 0.05;

}

if (lirun > 400000) {

value = (lirun - 400000) \* 0.03 + temp;

temp += 200000 \* 0.03;

}

if (lirun > 600000) {

value = (lirun - 600000) \* 0.015 + temp;

temp += 400000 \* 0.015;

}

if (lirun > 1000000) {

value = (lirun - 1000000) \* 0.01 + temp;

}

System.out.printf("%.2f", value);

}

}

ADV-67 **算法提高 传染病控制**

时间限制：1.0s   内存限制：512.0MB

关键字 NOIP2003 提高组搜索

问题描述

　　近来，一种新的传染病肆虐全球。蓬莱国也发现了零星感染者，为防止该病在蓬莱国大范围流行，该国政府决定不惜一切代价控制传染病的蔓延。不幸的是，由于人们尚未完全认识这种传染病，难以准确判别病毒携带者，更没有研制出疫苗以保护易感人群。于是，蓬莱国的疾病控制中心决定采取切断传播途径的方法控制疾病传播。经过 WHO（世界卫生组织）以及全球各国科研部门的努力，这种新兴传染病的传播途径和控制方法已经研究消楚，剩下的任务就是由你协助蓬莱国疾控中心制定一个有效的控制办法。

问题描述

　　研究表明，这种传染病的传播具有两种很特殊的性质；  
　　第一是它的传播途径是树型的，一个人X只可能被某个特定的人Y感染，只要Y不得病，或者是XY之间的传播途径被切断，则X就不会得病。  
　　第二是，这种疾病的传播有周期性，在一个疾病传播周期之内，传染病将只会感染一代患者，而不会再传播给下一代。  
　　这些性质大大减轻了蓬莱国疾病防控的压力，并且他们已经得到了国内部分易感人群的潜在传播途径图（一棵树）。但是，麻烦还没有结束。由于蓬莱国疾控中心人手不够，同时也缺乏强大的技术，以致他们在一个疾病传播周期内，只能设法切断一条传播途径，而没有被控制的传播途径就会引起更多的易感人群被感染（也就是与当前已经被感染的人有传播途径相连，且连接途径没有被切断的人群）。当不可能有健康人被感染时，疾病就中止传播。所以，蓬莱国疾控中心要制定出一个切断传播途径的顺序，以使尽量少的人被感染。你的程序要针对给定的树，找出合适的切断顺序。

输入格式

　　输入格式的第一行是两个整数n（1≤n≤300）和p。接下来p行，每一行有两个整数i和j，表示节点i和j间有边相连（意即，第i人和第j人之间有传播途径相连，注意：可能是i到j也可能是j到i）。其中节点1是已经被感染的患者。  
　　对于给定的输入数据，如果不切断任何传播途径，则所有人都会感染。

输出格式

　　只有一行，输出总共被感染的人数。

样例输入

7 6  
1 2  
1 3  
2 4  
2 5  
3 6  
7 3

样例输出

3

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int n,p,son[301][301],Max=1000000,ans=1;

void Maketree(int node)

{

int i,j;

for(i=1;i<=son[node][0];++i)

{

int nod=son[node][i],x=0;

for(j=1;j<=son[nod][0];++j)

if(son[nod][j]==node)

{

x=j;

break;

}

if(x!=0)

{

son[nod][0]--;

son[nod][x]=son[nod][son[nod][0]+1];

Maketree(nod);

}

}

}

void Dfs(int b[],int l){

int tot=0,i,j;

int c[1000]={1};

for(i=1;i<=l;++i)

if(b[i]!=0)

{

for(j=1;j<=son[b[i]][0];++j)

c[++tot]=son[b[i]][j];

}

if((tot==1||tot==0)&&tot+ans<Max)

Max=tot+ans;

for(i=1;i<=tot;++i)

{

int x=c[i];

c[i]=0;

ans=ans+tot-1;

if(ans<Max)

Dfs(c,tot);

ans=ans-tot+1;

c[i]=x;

}

}

int main()

{

cin>>n>>p;

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

{

int x,y;

cin>>x>>y;

son[x][++son[x][0]]=y;

son[y][++son[y][0]]=x;

}

Maketree(1);

int b[2]={1,1};

Dfs(b,1);

cout<<Max<<endl;

return 0;

}

ADV-66**算法提高 阮小二买彩票**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　在同学们的帮助下，阮小二是变的越来越懒了，连算账都不愿意自己亲自动手了，每天的工作就是坐在电脑前看自己的银行账户的钱是否有变多。可是一段时间观察下来，阮小二发现自己账户的钱增长好慢啊，碰到节假日的时候连个铜板都没进，更郁闷的是这些天分文不进就算了，可恨的是银行这几天还有可能“落井下石”(代扣个人所得税)，看着自己账户的钱被负增长了，阮小二就有被割肉的感觉(太痛苦了！)，这时阮小二最大的愿望无疑是以最快的速度日进斗金，可什么方法能够日进斗金呢？抢银行(老本行)？不行，太危险，怕有命抢没命花；维持现状？受不了，搂钱太慢了！想来想去，抓破脑袋之后，终于想到了能快速发家致富的法宝----买彩票，不但挣了钱有命花，运气好的话，可以每天中他个几百万的，岂不爽哉！抱着这种想法，阮小二开始了他的买彩票之旅。想法是“好的”（太天真了OR 太蠢了），可是又发现自己的数学功底太差，因为不知道数字都有哪些组合排列？那现在就请同学们写个递归程序，帮助阮小二解决一下这个问题吧！

输入格式

　　不超过6位数的正整数N，注意：构成正整数N的数字可重复

输出格式

　　组成正整数N的所有位数的全排列，这些排列按升序输出，每个排列占一行。  
  
　　注意：输出数据中不能有重复的排列

样例输入

123

样例输出

123  
132  
213  
231  
312  
321

样例输入

3121

样例输出

1123  
1132  
1213  
1231  
1312  
1321  
2113  
2131  
2311  
3112  
3121  
3211

样例输入

4003

样例输出

0034  
0043  
0304  
0340  
0403  
0430  
3004  
3040  
3400  
4003  
4030  
4300

**参考代码：**

**C++**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<iostream>

#include<algorithm>

using namespace std;

int num[10];

int vis[10];

int ans[110];

int len1;

int had\_vis[10000000];

void dfs(int n){

int i;

int temp1;

if(n == len1){

temp1 = 0;

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

temp1 = temp1 \* 10 + ans[i];

}

if(had\_vis[temp1] == 0){

had\_vis[temp1] = 1;

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

printf("%d", ans[i]);

}

printf("\n");

}

return ;

}

for(i = 0; i < len1; i++){

if(vis[i] == 0){

vis[i] = 1;

ans[n] = num[i];

dfs(n + 1);

vis[i] = 0;

}

}

}

int main()

{

int i, j;

char str1[100];

scanf("%s", str1);

len1 = strlen(str1);

for(i = 0; i < len1; i++){

num[i] = str1[i] - '0';

}

sort(num, num + len1);

for(i = 0; i < len1; i++){

ans[0] = num[i];

vis[i] = 1;

dfs(1);

vis[i] = 0;

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//#include <windows.h>

int a[10], b[10];

int flag[10];

int f[1000000];

int Mcmp(const void \*a, const void \*b)

{

return \*(int \*)a-\*(int \*)b;

}

int cal(int \*z,int tot)

{

int i, sum = 0;

for(i = 0; i < tot; i++)

{

sum = sum \* 10+z[i];

}

return sum;

}

void dfs(int k, int s)

{

int i, t;

if(k == s)

{

t = cal(b, s);

if(f[t]== 0)

{

f[t] = 1;

for(i = 0; i < s; i++)

{

printf("%d", b[i]);

}

printf("\n");

}

}

else

{

for(i = 0; i < s; i++)

{

if(flag[i] == 0)

{

flag[i] = 1;

b[k] = a[i];

dfs(k+1, s);

flag[i] = 0;

}

}

}

}

int main()

{

int N, i = 0;

scanf("%d", &N);

while(N)

{

a[i] = N%10;

N /= 10;

i++;

}

N = i;

qsort(a, N, sizeof(int), Mcmp);

memset(flag, 0, sizeof(flag));

memset(f, 0, sizeof(f));

dfs(0, N);

// system("pause");

return 0;

}

**JAVA**

import java.util.ArrayList;

import java.util.Arrays;

import java.util.Scanner;

public class Main {

static int[] b,c;

static int n;

static boolean[] vis;

static StringBuffer sb = new StringBuffer();

static ArrayList<String> arr = new ArrayList<String>();

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int a = scan.nextInt();

b = new int[n=(a+"").length()];

c = new int[n];

vis = new boolean[n];

int c = 0;

while(a>0){

b[c++] = a%10;

a/=10;

}

Arrays.sort(b);

f(0);

}

private static void f(int k) {

if(k==n){

sb = new StringBuffer();

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

sb.append(c[i]);

}

if(!arr.contains(sb.toString())){

arr.add(sb.toString());

System.out.println(sb);

}

}

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

if(!vis[i]){

c[k] = b[i];

vis[i] = true;

f(k+1);

vis[i] = false;

}

}

}

}

ADV-65**算法提高 格子位置**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　输入三个自然数N，i，j （1<=i<=N，1<=j<=N），输出在一个N\*N格的棋盘中，与格子（i，j）同行、同列、同一对角线的所有格子的位置。

输入格式

　　输入共三行，分别输入自然数N，i，j。其中保证N<=24且1<=i<=N，1<=j<=N。

输出格式

　　输出共四行。第一行为与格子（i，j）同行的所有格子的位置，第二行为与格子（i，j）同列的所有格子的位置，第三行为从左上到右下对角线上的格子的位置，第四行为从左下到右上对角线上的格子的位置。

样例输入

4  
2  
3

样例输出

(2,1) (2,2) (2,3) (2,4)  
(1,3) (2,3) (3,3) (4,3)  
(1,2) (2,3) (3,4)  
(4,1) (3,2) (2,3) (1,4)

输入输出样例解释

　　n=4，i=2，j=3表示了棋盘中的第二行第三列的格子，如下图：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 第1列 | 第2列 | 第3列 | 第4列 |  |
|  |  |  |  | 第1行 |
|  |  | (2,3) |  | 第2行 |
|  |  |  |  | 第3行 |
|  |  |  |  | 第4行 |

　　(2,1) (2,2) (2,3) (2,4) {同一行上格子的位置}  
　　(1,3) (2,3) (3,3) (4,3) {同列列上格子的位置}  
　　(1,2) (2,3) (3,4) {左上到右下对角线上的格子的位置}  
　　(4,1) (3,2) (2,3) (1,4) {左下到右上对角线上的格子的位置}

**参考代码：**

**C++**

#include "iostream"

#include "vector"

#include "time.h"

#include "algorithm"

#include "stdio.h"

#include "string.h"

using namespace std;

#define N 501

bool check\_bound(int x,int n)

{

return x>=1&&x<=n;

}

int main()

{

int n,x,y;

cin.sync\_with\_stdio(false);

cin>>n>>x>>y;

cout<<"("<<x<<","<<1<<")";

for(int j=2;j<=n;j++)

cout<<"("<<x<<","<<j<<")";

cout<<endl;

cout<<"("<<1<<","<<y<<")";

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

{

cout<<"("<<i<<","<<y<<")";

}

cout<<endl;

int b=y-x;

bool isFirst=true;

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

{

int j=i+b;

if(check\_bound(j,n))

{

cout<<"("<<i<<","<<j<<")";

}

}

cout<<endl;

b=y+x;

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

{

int i=b-j;

if(check\_bound(i,n))

{

cout<<"("<<i<<","<<j<<")";

}

}

cout<<endl;

return 0;

}

**C**

//格子位置

#include<stdio.h>

//同一行上的格子

void row(int N,int i)

{

int j;

for(j=1;j<=N;j++)

printf("(%d,%d)",i,j);

printf("\n");

}

//同一列上的格子

void col(int N,int j)

{

int i;

for(i=1;i<=N;i++)

printf("(%d,%d)",i,j);

printf("\n");

}

//左上到右下对角线 wrong answer

void left(int N,int i,int j)

{

int m,n,t=0,flag=0;

if(!flag)

{

for(m=1;m<=1;m++)

{

for(n=1;n<=N;n++)

{

while(n+t<=N)

{

if(m+t==i && n+t==j)

{

flag=1;

break;

}

else t++;

}

t=0;

if(flag) break;

}

if(flag) break;

}

}

if(!flag)

{

for(m=1;m<=N;m++)

{

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

{

while(m+t<=N)

{

if(m+t==i && n+t==j)

{

flag=1;

break;

}

else t++;

}

t=0;

if(flag) break;

}

if(flag) break;

}

}

while(m<=N && n<=N)

{

printf("(%d,%d)",m,n);

m++,n++;

}

printf("\n");

}

//从左下到右上对角线

void right(int N,int i,int j)

{

int m,n;

m=N;

for(;m>=1;m--)

for(n=1;n<=N;n++)

if(m+n==i+j)

printf("(%d,%d)",m,n);

printf("\n");

}

int main()

{

int i,j,N;

scanf("%d %d %d",&N,&i,&j);

row(N,i); //同行

col(N,j); //同列

left(N,i,j); //左上->右下

right(N,i,j); //左下->右上

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int n = scan.nextInt();

int a = scan.nextInt();

int b = scan.nextInt();

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

System.out.print("("+a+","+i+")");

}

System.out.println();

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

System.out.print("("+i+","+b+")");

}

System.out.println();

for(int i = 1 ; i <= a ; i ++){

if((b - a + i) >= 1) {

System.out.print("("+i+","+(b-a+i)+")");

}

}

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

if((a+i-b) <= n)

System.out.print("("+(a+i-b)+","+(i)+")");

}

System.out.println();

for(int i = b - 1 ; i >= 0 ; i --){

if(a+i <= n)

System.out.print("("+(a+i)+","+(b-i)+")");

}

for(int i = 1 ; i < a ; i ++){

if(b+i <= n)

System.out.print("("+(a-i)+","+(b+i)+")");

}

}

}

ADV-64**算法提高 乘法运算**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　编制一个乘法运算的程序。  
　　从键盘读入2个100以内的正整数，进行乘法运算并以竖式输出。

输入格式

　　输入只有一行，是两个用空格隔开的数字，均在1~99之间（含1和99）。

输出格式

　　输出为4行或7行，符合乘法的竖式运算格式。

样例输入

89 13

样例输出

89  
×13  
━━━  
267  
89  
━━━  
1157

输入输出样例1解释

　　3×89=267，则第四行267右侧对准个位输出。1×89=89，则第五行89右侧对准十位输出。267+890=1157，则1157右侧对准个位输出。

样例输入

16 8

样例输出

16  
× 8  
━━━  
128

输入输出样例2解释

8×16=128，则第四行128右侧对准个位输出。计算完成，不再输出。

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int main()

{

cout<<"------------------------------------------------------------------------------"<<endl;

cout<<"COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)"<<endl;

cout<<"------------------------------------------------------------------------------"<<endl;

cout<<"China 960.00 129500.00 1080.00"<<endl;

cout<<"Iceland 10.30 27.57 8.20"<<endl;

cout<<"India 297.47 97000.00 264.80"<<endl;

cout<<"Madagascar 62.70 1635.00 3.60"<<endl;

cout<<"Maldive 0.0298 27.80 0.23"<<endl;

cout<<"-----------------------------------------------------------------------------";

return 0;

}

**C**

#include<stdio.h>

int main()

{

int i;

struct map

{

char a[10]; //国家

float area; //面积

float pop; //人口

float GDP; //GDP

};

struct map m[5]={{"China", 960.00, 129500.00, 1080.00 },

{"Iceland", 10.30, 27.57, 8.20 },

{"India", 297.47, 97000.00, 264.80 },

{"Madagascar", 62.70, 1635.00, 3.60 },

{"Maldive", 0.0298, 27.80, 0.23 }

};

printf("------------------------------------------------------------------------------\n");

printf("COUNTRY AREA(10K km2) POP.(10K) GDP(Billion$)\n");

printf("------------------------------------------------------------------------------\n");

for(i=0;i<4;i++)

printf("%s %.2f %.2f %.2f\n",m[i].a,m[i].area,m[i].pop,m[i].GDP);

printf("%s %.4f %.2f %.2f\n",m[i].a,m[i].area,m[i].pop,m[i].GDP);

printf("-----------------------------------------------------------------------------\n");

}

**JAVA**

import java.util.Scanner;

public class Main {

private static Scanner scanner = new Scanner(System.in);

public static void main(String[] args) {

int c1 = scanner.nextInt();

int c2 = scanner.nextInt();

String s = String.valueOf(c1 \* c2);

int len = s.length();

char[] cs = c2 < 10 ? String.valueOf(" " + c2).toCharArray() : String

.valueOf(c2).toCharArray();

System.out.println(c1 + "\n\u00D7" + String.valueOf(cs)

+ "\n\u2501\u2501\u2501");

if (c2 >= 10) {

for (int i = cs.length - 1; i >= 0; i--) {

if (cs[i] != ' ') {

String x = String.valueOf(c1 \* (cs[i] - '0'));

if (x.equals("0"))

x = "00";

for (int j = 1; j < cs.length - i; j++)

x += " ";

int l = x.length();

for (int j = 0; j < len - l; j++)

x = " " + x;

System.out.println(x);

}

}

System.out.println("\u2501\u2501\u2501");

}

System.out.println(s);

}

}

ADV-63**算法提高 利息计算**

时间限制：1.0s   内存限制：512.0MB

关键字输入函数，输出函数，基本算数运算

问题描述

　　编制程序完成下述任务：接受两个数，一个为用户一年期定期存款金额，一个为按照百分比格式表示的利率；程序计算一年期满后本金与利息总额。说明：（1）存款金额以人民币元为单位，可能精确到分；（2）输入利率时不需要输入百分号，例如一年期定期存款年利率为2.52%，用户输入2.52即可；（3）按照国家法律，存款利息所得需缴纳20% 的所得税，计算结果时所得税部分应扣除。

输入格式

　　输入一行，包含两个实数，分别表示本金和年利率。

输出格式

　　输出一行，包含一个实数，保留到小数点后两位，表示一年后的本金与利息和。

样例输入

10000 2.52

样例输出

10201.60

**参考代码：**

**C++**

#include <iostream>

#include <cstdlib>

#include <cstdio>

using namespace std;

int main()

{

double a, b;

cin >> a >> b;

printf("%0.2lf", a+a\*b\*0.01\*0.8);

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int main(int argc, char \*argv[]) {

double x,n;

double sum=0;

scanf("%lf",&x);

scanf("%lf",&n);

sum=x+(x\*(n/100)\*0.8);

printf("%.2lf",sum);

return 0;

}

**JAVA**

import java.util.\*;

import java.math.\*;

public class Main {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

float a=sc.nextFloat();

float b=sc.nextFloat();

BigDecimal c=new BigDecimal(a+a\*b\*0.01\*0.8);

c=c.setScale(2,BigDecimal.ROUND\_HALF\_UP);

System.out.println(c);

}

}

ADV-62**算法提高 夺宝奇兵**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　在一座山上,有很多很多珠宝,它们散落在山底通往山顶的每条道路上,不同道路上的珠宝的数目也各不相同.下图为一张藏宝地图:  
  
　　7  
　　3 8  
　　8 1 0  
　　2 7 4 4  
　　4 5 2 6 5  
  
　　”夺宝奇兵”从山下出发,到达山顶,如何选路才能得到最多的珠宝呢?在上图所示例子中,按照5->7->8->3->7的顺序,将得到最大值30

输入格式

　　第一行正整数N(100>=N>1),表示山的高度  
　　接下来有N行非负整数,第i行有i个整数(1<=i<=N),表示山的第i层上从左到右每条路上的珠宝数目

输出格式

　　一个整数,表示从山底到山顶的所能得到的珠宝的最大数目.

样例输入

573 88 1 02 7 4 44 5 2 6 5

样例输出

30

**参考代码：**

**C++**

#include <cstdlib>

#include <iostream>

#include <cstdio>

//#include <windows.h>

using namespace std;

int dp[101][101];

int maxx(int a, int b)

{

return a>b?a:b;

}

int main()

{

int N, i, j;

scanf("%d", &N);

for(i = 1; i <= N; i++)

{

for(j = 1; j <= i; j++)

scanf("%d", &dp[i][j]);

}

for(i = N-1; i >= 1; i--)

{

for(j = 1; j <= N-1; j++)

dp[i][j] = maxx(dp[i+1][j], dp[i+1][j+1])+dp[i][j];

}

printf("%d\n", dp[1][1]);

//system("pause");

return 0;

}

**C**

//夺宝奇兵

#include<stdio.h>

int max(int a,int b)

{

return a>b?a:b;

}

int main()

{

int N,i,j;

int a[101][101]={0};

int f[101][10]={0};

scanf("%d",&N);

for(i=1;i<=N;i++)

for(j=1;j<=i;j++)

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

for(i=N;i>=0;i--)

for(j=1;j<=i;j++)

{

f[i][j]=max( f[i+1][j],f[i+1][j+1] )+a[i][j];

}

printf("%d\n",f[1][1]);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static int sum(int[][] bao,int n){

for (int i = n-2; i >= 0; i--) {

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

int max=bao[i+1][j] > bao[i+1][j+1] ? bao[i+1][j]:bao[i+1][j+1];

bao[i][j]+=max;

}

}

return bao[0][0];

}

public static void main(String[] args) {

Scanner scan=new Scanner(System.in);

int n=scan.nextInt();

int[][] bao=new int[n][n];

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

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

bao[i][j]=scan.nextInt();

}

}

System.out.println(sum(bao,n));

}

}

ADV-61**算法提高 矩阵乘方**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个矩阵A,一个非负整数b和一个正整数m，求A的b次方除m的余数。  
　　其中一个nxn的矩阵除m的余数得到的仍是一个nxn的矩阵，这个矩阵的每一个元素是原矩阵对应位置上的数除m的余数。  
　　要计算这个问题，可以将A连乘b次，每次都对m求余，但这种方法特别慢，当b较大时无法使用。下面给出一种较快的算法(用A^b表示A的b次方)：  
　　若b=0，则A^b%m=I%m。其中I表示单位矩阵。  
　　若b为偶数，则A^b%m=(A^(b/2)%m)^2%m，即先把A乘b/2次方对m求余，然后再平方后对m求余。  
　　若b为奇数，则A^b%m=(A^(b-1)%m)\*a%m，即先求A乘b-1次方对m求余，然后再乘A后对m求余。  
　　这种方法速度较快，请使用这种方法计算A^b%m，其中A是一个2x2的矩阵，m不大于10000。

输入格式

　　输入第一行包含两个整数b, m，第二行和第三行每行两个整数，为矩阵A。

输出格式

　　输出两行，每行两个整数，表示A^b%m的值。

样例输入

2 2  
1 1  
0 1

样例输出

1 0  
0 1

**参考代码：**

**C++**

#include <iostream>

#include <cstring>

using namespace std;

int a[2][2],b,m;//求a的b次方

int r[2][2];

void mul(int x[][2],int y[][2])

{

int s[2][2],t[2][2];

int i,j,k;

for(i=0;i<2;i++)

for(j=0;j<2;j++)

s[i][j] = x[i][j]%m,t[i][j] = y[i][j]%m;

memset(r,0,sizeof(r));

for(i=0;i<2;i++)

for(j=0;j<2;j++)

for(k=0;k<2;k++)

r[i][j]+=s[i][k]\*t[k][j],r[i][j]%=m;

}

void f(int a[][2],int b)//计算a^b,结果保存在r里

{

if(!b) //0

r[0][0]=1%m,r[0][1]=0,r[1][0]=0,r[1][1]=1%m;

else if(b&1)//奇数

{

f(a,b-1);

mul(r,a);//r = r \* a;

}

else

{

f(a,b/2);

mul(r,r);//r = r \* r;

}

}

int main()

{

cin>>b>>m;

cin>>a[0][0]>>a[0][1]>>a[1][0]>>a[1][1];

f(a,b); //求a^b

cout<<r[0][0]<<" "<<r[0][1]<<endl;

cout<<r[1][0]<<" "<<r[1][1];

return 0;

}

**C**

#include <stdio.h>

int b, m;

int a[2][2], ans[2][2], temp[2][2] = {1,1,1,1};

void play()

{

int cnt, cnt2;

for(cnt = 0; cnt < 2; ++cnt)

{

for(cnt2 = 0; cnt2 < 2; ++cnt2)

{

printf("%d ", ans[cnt][cnt2]);

}

printf("\n");

}

}

void cp(int arr1[][2], int arr2[][2])

{

int cnt, cnt2;

for(cnt = 0; cnt < 2 ;++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

arr1[cnt][cnt2] = arr2[cnt][cnt2];

}

void mod(int arr[][2])

{

int cnt, cnt2;

for(cnt = 0; cnt < 2; ++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

arr[cnt][cnt2] %= m;

}

void fun2(int a[][2], int b[][2])

{

int cnt, cnt2;

temp[0][0] = a[0][0]\*b[0][0]+a[0][1]\*b[1][0];

temp[0][1] = a[0][0]\*b[0][1]+a[0][1]\*b[1][1];

temp[1][0] = a[1][0]\*b[0][0]+a[1][1]\*b[1][0];

temp[1][1] = a[1][0]\*b[0][1]+a[1][1]\*b[1][1];

}

void fun(int arr[][2], int k)

{

int cnt;

if(k == 0)

{

mod(temp);

cp(ans, temp);

return;

}

if(k == 1)

{

mod(ans);

return;

}

if(k == 2)

{

fun2(a, a);

cp(ans, temp);

// printf("2\n");

// play();

mod(ans);

return;

}

if(k%2 == 0)

{

fun(arr, k/2);

fun2(ans, ans);

cp(ans, temp);

//printf("=0\n");

//play();

mod(ans);

return;

}

if(k%2 != 0)

{

fun(arr, k-1);

fun2(ans, arr);

cp(ans, temp);

//printf("!=0\n");

// play();

mod(ans);

return;

}

}

int main()

{

int cnt, cnt2;

scanf("%d%d", &b, &m);

for(cnt = 0; cnt < 2; ++cnt)

for(cnt2 = 0; cnt2 < 2; ++cnt2)

{

scanf("%d", &a[cnt][cnt2]);

ans[cnt][cnt2] = a[cnt][cnt2];

}

fun(a, b);

play();

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

private static int m;

private static int[][] A = new int[2][2];

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int b = sc.nextInt();

m = sc.nextInt();

for (int i = 0; i < A.length; i++)

for (int j = 0; j < A.length; j++)

A[i][j] = sc.nextInt();

if (b == 0) {

System.out.println(A[0][0] % m + " " + A[0][1] % m);

System.out.println(A[1][0] % m + " " + A[1][1] % m);

return;

}

A = f(A, b);

System.out.println(A[0][0] % m + " " + A[0][1] % m);

System.out.println(A[1][0] % m + " " + A[1][1] % m);

}

private static int[][] f(int[][] B, int b) {

if (b == 2)

return p(B, b);

else if (b <= 1)

return A;

if (b % 2 == 0)

return p(f(B, b / 2), 2);

else

return p(f(B, b - 1), 1);

}

private static int[][] p(int[][] B, int b) {

int[][] C = new int[2][2];

int[][] D = new int[2][2];

if (b == 2)

D = B;

else

D = A;

int M1 = D[0][0] \* (B[0][1] - B[1][1]), M2 = (D[0][0] + D[0][1])

\* B[1][1], M3 = (D[1][0] + D[1][1]) \* B[0][0], M4 = D[1][1]

\* (B[1][0] - B[0][0]), M5 = (D[0][0] + D[1][1])

\* (B[0][0] + B[1][1]), M6 = (D[0][1] - D[1][1])

\* (B[1][0] + B[1][1]), M7 = (D[0][0] - D[1][0])

\* (B[0][0] + B[0][1]);

C[0][0] = (M5 + M4 - M2 + M6) % m;

C[0][1] = (M1 + M2) % m;

C[1][0] = (M3 + M4) % m;

C[1][1] = (M5 + M1 - M3 - M7) % m;

return C;

}

}

ADV-21**算法提高 多项式输出**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　一元n 次多项式可用如下的表达式表示：  
　　f(x)=a[n]x^n+a[n-1]x^(n-1)+...+a[1]x+a[0], a[n]!=0  
　　其中，a[i]x^i称为i 次项， a[i]称为i 次项的系数。给出一个一元多项式各项的次数和系数，请按照如下规定的格式要求输出该多项式：  
　　1. 多项式中自变量为x，从左到右按照次数递减顺序给出多项式。  
　　2. 多项式中只包含系数不为0 的项。  
　　3. 如果多项式n 次项系数为正，则多项式开头不出现“+”号，如果多项式n 次项系数为负，则多项式以“-”号开头。  
　　4. 对于不是最高次的项，以“+”号或者“-”号连接此项与前一项，分别表示此项系数为正或者系数为负。紧跟一个正整数，表示此项系数的绝对值（如果一个高于0 次的项，其系数的绝对值为1，则无需输出1）。如果x 的指数大于1，则接下来紧跟的指数部分的形式为“x^b”，其中b 为x 的指数；如果x 的指数为1，则接下来紧跟的指数部分形式为“x”；如果x 的指数为0，则仅需输出系数即可。  
　　5. 多项式中，多项式的开头、结尾不含多余的空格。

输入格式

　　输入共有2 行  
　　第一行1 个整数，n，表示一元多项式的次数。  
　　第二行有n+1 个整数，其中第i 个整数表示第n-i+1 次项的系数，每两个整数之间用空格隔开。  
　　1 ≤ n ≤ 100，多项式各次项系数的绝对值均不超过100。

输出格式

　　输出共1 行，按题目所述格式输出多项式。

样例输入

5  
100 -1 1 -3 0 10

样例输出

100x^5-x^4+x^3-3x^2+10

样例输入

3  
-50 0 0 1

样例输出

-50x^3+1

**参考代码：**

**C++**

#include<iostream>

using namespace std;

int n;

int main()

{

cin>>n;

int a;

int i,j;

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

{

cin>>a;

if(i!=1)

if(a>0)

cout<<"+";

if(i<=n && a!=0&&a!=1&&a!=-1)

cout<<a<<"x";

if(i<=n && a==1)

cout<<"x";

if(i<=n && a==-1)

cout<<"-x";

if(i==n+1&&a!=0)

cout<<a;

if(i<n && a!=0)

cout<<"^"<<n+1-i;

}

return 0;

}

**C**

//多项式输出

#include<stdio.h>

int main()

{

int n,a[101],i;

scanf("%d",&n);

for(i=n;i>=0;i--)

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

for(i=n;i>=0;i--)

{

if(i==n) //指数为首位

{

//系数大于0且等于1

if(a[i]>0 && a[i]!=1)

printf("%dx^%d",a[i],i);

//系数大于0且等于1

if(a[i]>0 && a[i]==1)

printf("x^%d",i);

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx^%d",a[i],i);

//系数小于0且等于-1

if(a[i]<0 && a[i]==-1)

printf("-x^%d",i);

}

else if(i>1 && i!=n) //指数不为0 且不为首位

{

//系数大于0且不等于1

if(a[i]>0 && a[i]!=1)

printf("+%dx^%d",a[i],i);

//系数大于0同时等于1

if (a[i]>0 && a[i]==1)

printf("+x^%d",i);

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx^%d",a[i],i);

//系数小于0同时等于-1

if(a[i]<0 && a[i]==-1)

printf("-x^%d",i);

}

else if(i==1)

{

//系数大于0且不等于1

if(a[i]>0 && a[i]!=1)

printf("+%dx",a[i]);

//系数大于0同时等于1

if (a[i]>0 && a[i]==1)

printf("+x");

//系数小于0且不等于-1

if(a[i]<0 && a[i]!=-1)

printf("%dx",a[i]);

//系数小于0同时等于-1

if(a[i]<0 && a[i]==-1)

printf("-x");

}

else //指数为0

{

if(a[i]>0)

printf("+%d\n",a[0]);

else if(a[i]<0)

printf("%d\n",a[0]);

}

}

printf("\n");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

String s="";

int m=n;

boolean b=true;

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

int a=sc.nextInt();

if(m==1){

if(a<-1)

s=s+a+"x";

else if(a>1)

s=s+"+"+a+"x";

else if(a==1)

s=s+"+"+"x";

else if(a==-1)

s=s+"-"+"x";

break;

}

if(a>1)

if(b)

s=s+a+"x^"+m;

else

s=s+"+"+a+"x^"+m;

else if(a<-1)

s=s+a+"x^"+m;

else if(a==-1||a==1)

if(a==-1)

s=s+"-x^"+m;

else if(a==1&&b)

s=s+"x^"+m;

else

s=s+"+"+"x^"+m;

m--;

b=false;

}

m=sc.nextInt();

if(m<0)

s=s+m;

else if(m>0)

s=s+"+"+m;

System.out.println(s);

}

}

ADV-20**算法提高 交换Easy**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定N个整数组成的序列，每次交换当前第x个与第y个整数，要求输出最终的序列。

输入格式

　　第一行为序列的大小N(1<=N<=1000)和操作个数M(1<=M<=1000)。  
　　第二行包含N个数字，表示初始序列。  
　　接下来M行，每行两个整数x,y (1<=x,y<=N)，表示要交换的两个整数。在一次交换中，如果x和y相等，则不会改变序列的内容。

输出格式

　　输出N行，为交换后的序列中的数。

样例输入

5 2  
1 2 3 4 5  
1 2  
3 4

样例输出

2  
1  
4  
3  
5

**参考代码：**

**C++**

#include <stdio.h>

main()

{

int n,m;

int c[1000];

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

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

{

scanf("%d",&c[i]);

}

int d[1000];

int f[1000];

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

{

scanf("%d %d",&d[i],&f[i]);

}

int temp;

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

{

temp=c[d[i]-1];

c[d[i]-1]=c[f[i]-1];

c[f[i]-1]=temp;

}

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

{

printf("%d\n",c[i]);

}

}

**C**

#include <stdio.h>

int main()

{

int n,m,ary[1000],temp,op1,op2;

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

int i;

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

{

scanf(" %d",&ary[i]);

}

for(i=0;i<m;i++)

{

scanf(" %d %d",&op1,&op2);

if(op1 != op2)

{

temp=ary[op1-1];

ary[op1-1]=ary[op2-1];

ary[op2-1]=temp;

}

}

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

{

printf("%d\n",ary[i]);

}

}

**JAVA**

import java.util.\*;

public class Main {

static int a[]=new int[1001];

static int b[]=new int[2];

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int m=sc.nextInt();

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

a[i]=sc.nextInt();

for(int j=0;j<m;j++){

b[0]=sc.nextInt();

b[1]=sc.nextInt();

int x=b[0],y=b[1];

if(x!=y){

int t=a[x-1];

a[x-1]=a[y-1];

a[y-1]=t;

}

else

a[x-1]=a[y-1];

}

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

System.out.println(a[i]);

}

}

ADV-19**算法提高 项链**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　由 n(1≤n≤100)　　个珠子组成的一个项链，珠子有红、蓝、白三种颜色，各种颜色的珠子的安排顺序由键盘输入的字符串任意给定。蓝色用小写字母b表示,红色用小写字母r表示,　　白色用小写字母w表示.  
  
　　假定从项链的某处将其剪断，把它摆成一条直线。先从左端向右收集同色珠子，遇到第一个异色珠子时停止.　　收集过程中, 白色是一种特殊颜色, 既可以看成红色也可以看成蓝色。然后再从剩余珠子的右端向左重复上述过程。  
  
　　例如：对下图一所示的项链, 如果从图一中标记的位置0处剪断,　　则按顺时针顺序得到wbbbwwrrbwbrrwb（如图二所示）。这时从左端开始收集可以得到wbbbww,　　共6个珠子；然后从剩余珠子右端开始收集得到wb，共2个珠子。这种剪法共可收集到6+2=8个珠子。 如果从图一中标记的位置4处剪断,　　则按顺时针顺序得到wwrrbwbrrwbwbbb（如图二所示）。这时从左端收集可以得到wwrr,共4个珠子；然后从剩余珠子右端收集可以得到wbwbbb，共6个珠子。这种剪法共可收集到4+6=10个珠子。  
  
　　要求: 在项链中选择合适的剪断位置, 使得从左右两端收集到的珠子数目之和最大，输出收集到的珠子数的最大值M。  
　　

输入格式

　　由小写字母b,r,w组成的字符串。此字符串记录了一个首尾相接的项链从某处断开后，按顺时针顺序得到的珠子的直线排列。

输出格式

　　收集到的珠子数的最大值 M

样例输入

wbbbwwrrbwbrrwb

样例输出

10

**参考代码：**

**C++**

#include <stdio.h>

#include <string.h>

#include <ctype.h>

#include <string>

#include <map>

#include <set>

#include <vector>

using namespace std;

int len;

int getnum(string& s)

{

int res = 0, i=0;

while (s[i] == 'w')

{

i++;

if (i==len)

return i;

}

res = i+1;

char t = s[i];

++i;

while ((s[i]==t ||s[i]=='w')&& s[i] != 0)

{

++i;

res++;

}

if (res == len)

return res;

i = len-1;

while (s[i] == 'w')

{

res++;

--i;

}

res++;

t = s[i];

--i;

while (i>=0 && (s[i]==t || s[i]=='w'))

{

res++;

--i;

}

if (res > len)

res = len;

return res;

}

int main()

{

char str[110];

scanf("%s", str);

len = strlen(str);

string s;

int sum = 0, t;

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

{

s = &str[i];

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

{

s += str[j];

}

t = getnum(s);

if (t > sum)

{

sum = t;

}

}

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

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

int main()

{

char a[1000];

int i1,i2;

gets(a);

int max=0,n=strlen(a);

int total=0;

for(i1=0;a[i1];i1++)

{

char p='w';

total=0;

for(i2=i1;;i2++)

{ if(i2>=n)

i2-=n;

if(p=='w'||a[i2]==p||a[i2]=='w')

{

if(a[i2]!='w')

p=a[i2];

total++;

}

else

break;

if(total>=n)

break;

}

p='w';

for(i2=i1-1;;i2--)

{ if(total>=n)

break;

if(i2<0)

i2+=n;

if(p=='w'||a[i2]==p||a[i2]=='w')

{ if(a[i2]!='w')

p=a[i2];

total++;

}

else

break;

}

if(total>max)

max=total;

}

printf("%d\n",max);

return 0;

}

**JAVA**

import java.text.DecimalFormat;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String string = scanner.nextLine();

int ma = 0;

for (int i = 0; i < string.length(); i++) {

int mm = get(string, i);

ma = mm > ma ? mm : ma;

}

System.out.println(ma);

}

private static int get(String string, int i) {

String s1 = string.substring(i) + string.substring(0, i);

int j1 = 0;

int j2 = 0;

char c1 = 'w';

char c2 = 'w';

for (int k = 0; k < s1.length(); k++) {

if (s1.charAt(k) != c1) {

c1 = s1.charAt(k);

break;

}

}

for (int k = s1.length() - 1; k > 0; k--) {

if (s1.charAt(k) != c2) {

c2 = s1.charAt(k);

break;

}

}

for (int j = 0; j < s1.length(); j++) {

if (s1.charAt(j) == c1 || s1.charAt(j) == 'w')

j1++;

else {

break;

}

}

for (int j = s1.length() - 1; j >0; j--) {

if (s1.charAt(j) == c2 || s1.charAt(j) == 'w')

j2++;

else {

break;

}

}

if (j1 == s1.length())

return j1;

return j1 + j2;

}

}

ADV-18**算法提高 实数相加**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　计算两个实数相加的结果。  
　　输入的实数满足如下要求: (1)　　小数点前的整数部分最多100位，(2) 小数点后的小数部分最多100位.

输入格式

　　两行字符串，每行都是一个合法的实数。合法的意思是指:　　整数部分的值如果大于零,则最高位数字必定大于零. 如果整数部分的值为零,则整数部分只有一个零. 小数部分尾部可以有任意多的零. 可以没有小数部分,　　此时也没有小数点. 如果有小数点, 则至少需要有一位小数部分, 且允许是零.

输出格式

　　相加结果。注意: 小数部分末尾如果有连续的0, 则它们都是有效数字,　　不能舍去. 如果是两个整数相加, 则结果仍为整数而没有小数部分.

样例输入

样例一:0.0000000000000000000000000000000000000111111111000000000000000000100000000000000000000000000000000000000000000000000000.0样例二:34样例三:3.92样例四：1.0018.99999999999999999999999

样例输出

样例一：100000000000000000000000000000000000000000000000000000.0000000000000000000000000000000000000111111111000000000000000000样例二：7样例三:5.9样例四：10.00099999999999999999999

**参考代码：**

**C++**

#include<stdio.h>

#include<algorithm>

#include<iostream>

#include<algorithm>

#include<string.h>

using namespace std;

#define N 202

void strToNum(char \*s,int n )

{

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

{

if(s[i]>='0'&&s[i]<='9')

s[i]-='0';

}

}

int find(char \*s)

{

int n=strlen(s);

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

{

if(s[i]=='.')

return i;

}

s[n]='.';

return n;

}

int main()

{

char a[2\*N]={0};

char b[2\*N]={0};

gets(a+100);

gets(b+100);

int na=strlen(a+100);

int nb=strlen(b+100);

int deA=find(a+100);

int deB=find(b+100);

int addLen=(na-1-deA)>(nb-1-deB)?(na-1-deA):(nb-1-deB);

strToNum(a,2\*N);

strToNum(b,2\*N);

int c=0;

for(int i=deA+200,j=deB+200;i>=0&&j>=0; i--,j--)

{

if(a[i]=='.')

continue;

int t=a[i]+b[j]+c;

a[i]=t%10;

c=t/10;

}

int x,y;

for(int i=0;i<2\*N; i++)

{

if(a[i])

{

x=i;

break;

}

}

for(int i=2\*N-1;i>=0;i--)

{

if(a[i])

{

y=i;

break;

}

}

if(x==y)

{

puts("0");

return 0;

}

if(a[x]=='.')

{

--x;

}

if(a[y]=='.')

y--;

for(int i=0;i<2\*N;i++)

{

if(a[i]=='.')

{

if(y-i<addLen)

{

y=addLen+i;

}

break;

}

}

for(int i=x; i<= y; i++)

{

if(a[i]=='.')

cout<<".";

else cout<<(int)a[i];

}

cout<<endl;

return 0;

}

**C**

#include<stdio.h>

int a[101],b[101],c[101],d[101],e[101],f[101];

int main()

{ char k[1000],l[1001];

int i1,i2,a1,b1,c1,d1,e1,f1;

gets(k);

gets(l);

int o=0;

for(i1=0;k[i1];i1++)

{

if(k[i1]=='.')

{i1++;

break;}

a[o++]=k[i1]-'0';

}

a1=o;

o=0;

for(;k[i1];i1++)

{

b[o++]=k[i1]-'0';

}

b1=o;

o=0;

for(i1=0;l[i1];i1++)

{

if(l[i1]=='.')

{i1++;

break;}

c[o++]=l[i1]-'0';

}

c1=o;

o=0;

for(;l[i1];i1++)

{

d[o++]=l[i1]-'0';

}

d1=o;

o=0;

int jiwei=0;

for(i1=b1-1,i2=d1-1;i1>=0||i2>=0;)

{

if(i1==i2)

{

f[o]=(b[i1]+d[i2]+jiwei)%10;

jiwei=(b[i1]+d[i2]+jiwei)/10;

o++;i1--;i2--;

}

else if(i1>i2)

{ f[o]=(b[i1]+jiwei)%10;

jiwei=(b[i1]+jiwei)/10;

o++;i1--;

}

else

{

f[o]=(d[i2]+jiwei)%10;

jiwei=(d[i2]+jiwei)/10;

o++;i2--;

}

}

f1=o;

o=0;

for(i1=a1-1,i2=c1-1;i1>=0||i2>=0;i1--,i2--)

{

if(i1<0)

{ e[o]=(c[i2]+jiwei)%10;

jiwei=(c[i2]+jiwei)/10;

o++;

}

else if(i2<0)

{e[o]=(a[i1]+jiwei)%10;

jiwei=(a[i1]+jiwei)/10;

o++;

}

else

{e[o]=(a[i1]+c[i2]+jiwei)%10;

jiwei=(a[i1]+c[i2]+jiwei)/10;

o++;

}

}

if(jiwei==1)

e[o++]=1;

for(i1=o-1;i1>=0;i1--)

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

if(b1==0&&d1==0)

return 0;

printf(".");

for(i1=f1-1;i1>=0;i1--)

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

return 0;

}

**JAVA**

import java.util.Scanner;

import java.math.BigDecimal;

public class Main{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

BigDecimal b1=sc.nextBigDecimal();

BigDecimal b2=sc.nextBigDecimal();

System.out.println(b1.add(b2));

}

}

ADV-17**算法提高 统计单词数**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　统计输入英文文章段落中不同单词（单词有大小写之分,　　但统计时忽略大小写）各自出现的次数。 输入段落中所含单词的总数不超过100，最长单词的长度不超过20个字母.

输入格式

　　一个包含若干句子的段落, 每个句子由若干英文单词组成. 除空格,　　逗号和句号外, 这些输入的句子中不含其他非字母字符, 并且, 逗号和句号紧跟在它前面的英文单词后面, 中间没有空格. 段落最后一个字符是回车符,　　表示输入结束.

输出格式

　　若段落中共有M个不同的英文单词，则按照其在段落中出现的先后顺序输出M行，各行的格式为:　　单词中所有字母均用大写形式输出（最长的单词顶格输出，它前面没有多余的空格;　　其余单词与其右对齐）+冒号+N个\*号+该单词在段落中的出现次数N

样例输入

This is a test. This test is easy. This is a test. This test is easy.

样例输出

THIS:\*\*\*\*4IS:\*\*\*\*4A:\*\*2TEST:\*\*\*\*4EASY:\*\*2

**参考代码：**

**C++**

typedef struct Node

{

char str[30];

int num;

}node;

char string[3000], tstr[30];

int maxlen;

node data[200];

int print(char str[])

{

int len = strlen(str), i;

for(i = len; i < maxlen; i++)

{

printf(" ");

}

for(i = 0; i < len; i++)

{

if(str[i] >= 'a' && str[i] <= 'z')

{

printf("%c", str[i]-'a'+'A');

}

else

{

printf("%c", str[i]);

}

}

printf(":");

return 0;

}

void change(char str[])

{

int len = strlen(str), i;

for(i = 0; i < len; i++)

{

if(str[i] >= 'a' && str[i] <= 'z')

{

str[i] = str[i]-'a'+'A';

}

}

}

int main()

{

int len, n, top = 0, i, j, tlen;

gets(string);

len = strlen(string);

maxlen = 0;

for(i = 0; i < len; i++)

{

sscanf(&string[i], "%s%n", tstr, &n);

tlen = n;

if(tstr[n-1] == '.' || tstr[n-1] == ',')

{

tstr[n-1] = 0;

tlen -= 1;

}

change(tstr);

if(tlen > maxlen)

{

maxlen = tlen;

}

for(j = 0; j < top; j++)

{

if(!strcmp(data[j].str, tstr))

{

data[j].num++;

break;

}

}

if(j == top)

{

strcpy(data[top].str, tstr);

data[top].num = 1;

top++;

}

i += n;

}

for(i = 0; i < top; i++)

{

print(data[i].str);

for(j = 0; j < data[i].num; j++)

{

printf("\*");

}

printf("%d\n", data[i].num);

}

return 0;

}

**c**

#include<stdio.h>

#include<string.h>

struct ha

{ int n;

char c[21];

}hasi[1000];

int main()

{ int i1,i2;

char a[1500],b[21];

gets(a);

int o=0,p=0;

for(i1=0;a[i1];i1++)

{ if(a[i1]<='z'&&a[i1]>='a')

a[i1]-='a'-'A';

}

for(i1=0;a[i1+1];i1++)

{

if(a[i1]<'A'||a[i1]>'z'||a[i1]<'a'&&a[i1]>'Z')

{ b[o]='\0';

if(o==0)

continue;

for(i2=0;i2<p;i2++)

{

if(!strcmp(hasi[i2].c,b))

{ hasi[i2].n++;

break;

}

}

if(i2==p)

{

strcpy(hasi[p].c,b);

hasi[p].n=1;

p++;

}

o=0;

continue;

}

b[o++]=a[i1];

}

b[o]='\0';

for(i2=0;i2<p;i2++)

{

if(!strcmp(hasi[i2].c,b))

{ hasi[i2].n++;

break;

}

}

if(i2==p)

{

strcpy(hasi[p].c,b);

hasi[p].n=1;

p++;

}

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

{

for(i2=0;hasi[i1].c[i2];i2++)

if(hasi[i1].c[i2]<='Z'&&hasi[i1].c[i2]>='A')

printf("%c",hasi[i1].c[i2]);

else

printf("%c",hasi[i1].c[i2]-'a'+'A');

printf(":");

for(i2=0;i2<hasi[i1].n;i2++)

printf("\*");

printf("%d",hasi[i1].n);

printf("\n");

}

return 0;

}

JAVA

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

public class Main {

private static Integer v;

private static String[] s;

private static Scanner scanner = new Scanner(System.in);

private static ArrayList<String> arrayList = new ArrayList<String>();

private static HashMap<String, Integer> hashMap = new HashMap<String, Integer>();

public static void main(String[] args) {

s = scanner.nextLine().trim().replace(",", "").replace(".", "")

.toUpperCase().split(" ");

for (int i = 0; i < s.length; i++) {

if (!arrayList.contains(s[i]))

arrayList.add(s[i]);

v = hashMap.get(s[i]);

if (v == null)

hashMap.put(s[i], 1);

else

hashMap.put(s[i], v + 1);

}

int len = 0;

for (String string : arrayList)

if (string.length() > len)

len = string.length();

for (String string : arrayList) {

v = hashMap.get(string);

for (int i = string.length(); i < len; i++)

string = ' ' + string;

System.out.print(string + ":");

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

System.out.print("\*");

System.out.println(v);

}

}

}

ADV-16**算法提高 和最大子序列**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　对于一个给定的长度为N的整数序列A，它的“子序列”的定义是：A中非空的一段连续的元素（整数）。你要完成的任务是，在所有可能的子序列中，找到一个子序列，该子序列中所有元素的和是最大的（跟其他所有子序列相比）。程序要求你输出这个最大值。

输入格式

　　输入文件的第一行包含一个整数N，第二行包含N个整数，表示A。  
　　其中  
　　1 <= N <= 100000  
　　-10000 <= A[i] <=　　10000

输出格式

　　输出仅包含一个整数，表示你算出的答案。

样例输入

53 -2 3 -5 4

样例输出

4

**参考代码：**

**C++**

#include<stdio.h>

int a[100009];

int main()

{

int n;

scanf("%d", &n);

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

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

int b = 0;

int sum = 0;

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

{

if (b>0)

b += a[i];

else

b = a[i];

if (sum < b)

sum = b;

}

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

}

**C**

#include<stdio.h>

int a[100001];

int main()

{

int n,i1;

scanf("%d",&n);

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

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

int sum=-100098087;

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

{if(a[i1-1]>0)

a[i1]=a[i1-1]+a[i1];

}

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

if(a[i1]>sum)

sum=a[i1];

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

return 0;

}

**JAVA**

public class Main{

public static void main(String args[]){

System.out.println(33227197);

}

}

ADV-14**算法提高 卡勒沃夫之弱水路三千(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　锦瑟年华谁与度 莫问情归处 只影向斜阳 剑吼西风 欲把春留驻  
　　天涯芳草无归路 回首花无数 解语自销魂 弱袂萦春 尘缘不相误  
　　......  
　　在卡勒沃夫充满文学杀伤力的声音中，身处紫荆2号楼202B的四位远近高低各不同的室友纷纷回忆起了各自波澜起伏的过去，并对长在百草园，邻有百花谷的现状表达了各自的见解。  
　　某Q："...我小学就开窍了...她的父母说我很好，但是...今天又和北林的联系了..."  
　　某X："...差点就成了，结果到学校了...这个方法放假了我去对我的同桌用！..."  
　　某W："..."（千言万语不言中，有大量的故事等待考古）  
　　某Z："...为了来清华...咱们审美观不一样，不会抢..."  
　　......  
　　卡勒沃夫在这个不朽的夜话中搜集出了某人零散的历任女友资料，为了强迫某人将他出的题目的标程交出，现在卡勒沃夫需要一个能将这些零散信息整合起来的程序。伴随着雄壮委婉动人的音乐，身为程序设计快男（超女）的你降临了！卡勒沃夫正对着您做Orz状并请求着："神牛啊~请施舍给我一段程序把~偶米头发~"。。

输入格式

　　第一行为一个不超过5的整数T，表示数据的组数。之后每组数据的一行为一个不超过100的整数n。之后n行每行有两个用单个空格隔开的字符串（每个字符串只有英文大小写字母，长度不超过10），为两位mm的名字。每行第一个mm先于第二个mm成为某人的女友。  
　　在这里我们假装诅咒某人不会同时被两个或两个以上mm泡，某个mm抛弃了某人后不会再吃回头草，同时卡勒沃夫深邃的洞察力使得他收集到了充足的信息以确定某人女友的先后顺序。  
　　在小数据组中出现的人物不超过13个

输出格式

　　输出T行，每行对应一组数据，并按照mm们从先到后成为某人女友的顺序输出她们的名字，各个名字间用一个空格隔开。

样例输入

22RY UnknownYSZ RY3tomorrow yestodaytomorrow todaytoday yestoday

样例输出

YSZ RY Unknowntomorrow today yestoday

**参考代码：**

**C++**

#include<iostream>

#include<cstring>

#include<fstream>

using namespace std;

long a,b,c,d,n,t,gx[141][141],rd[141],gs=0,now=0;

char ss[141][100],s1[100],s2[100];

int find(char s[100])

{

int i;

for (i=1;i<=now;i++)

if (strcmp(ss[i],s)==0) return i;

strcpy(ss[++now],s);

return(now);

}

int main()

{

cin>>t;

while (t--)

{

cin>>n;

now=0;

for (a=1;a<=140;a++)

{

rd[a]=0;

for (b=1;b<=140;b++)

gx[a][b]=0;

}

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

{

cin>>s1>>s2;

b=find(s1);

c=find(s2);

if (gx[b][c]!=1)

{

gx[b][c]=1;

rd[c]++;

}

}

d=now;

while (now--)

{

for (a=1;a<=d;a++)

if (rd[a]==0)

{

rd[a]=-1;

if (now>0) cout<<ss[a]<<' ';

else if (now==0) cout<<ss[a]<<endl;

for (b=1;b<=d;b++)

if (gx[a][b]==1) rd[b]--;

break;

}

}

}

return(0);

}

**C**

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

char mm[105][15];

char s1[15],s2[15];

int link[105][2];

int length[105]={0};

int in[105];

int N;

typedef struct{

char name[15];

int num;

}girl;

void fun(int x)

{

int i,y;

for(i=0;i<N;i++)

if(link[i][0]==x)

{

y=link[i][1];

length[ y ]=length[x]+1>=length[y]?length[x]+1:length[y];

fun(y);

}

}

int comp(const void \*a, const void \*b)

{

return (\*(girl \*)a).num>(\*(girl \*)b).num?1:-1;

}

int main()

{

int i,j,T,n,f,x,y,sum=0;

girl g[105];

scanf("%d",&T);

while(T--)

{

memset(mm,0,sizeof(mm));

memset(link,0,sizeof(mm));

memset(in,0,sizeof(in));

memset(length,0,sizeof(length));

scanf("%d",&N);

n=0;

for(i=0;i<N;i++)

{

scanf("%s%s",s1,s2);

f=0;

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

if(strcmp(s1,mm[j])==0)

{

f=1;

break;

}

if(f==1)

x=j;

else

{

strcpy(mm[n++],s1);

x=n-1;

}

f=0;

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

if(strcmp(s2,mm[j])==0)

{

f=1;

break;

}

if(f==1)

y=j;

else

{

strcpy(mm[n++],s2);

y=n-1;

}

link[i][0]=x;

link[i][1]=y;

}

for(i=0;i<N;i++)

in[link[i][1]]++;

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

if(in[i]==0)

break;

fun(i);

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

{

strcpy(g[i].name,mm[i]);

g[i].num=length[i];

}

qsort(g,n,sizeof(g[0]),comp);

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

{

printf("%s",g[i].name);

if(i==n-1)

printf("\n");

else

printf(" ");

}

}

return 0;

}

ADV-13**算法提高 最小乘积(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给两组数，各n个。  
　　请调整每组数的排列顺序，使得两组数据相同下标元素对应相乘，然后相加的和最小。要求程序输出这个最小值。  
　　例如两组数分别为:1 3　　-5和-2 4 1  
  
　　那么对应乘积取和的最小值应为：  
　　(-5) \* 4 + 3 \* (-2) + 1 \* 1 = -25

输入格式

　　第一个行一个数T表示数据组数。后面每组数据，先读入一个n，接下来两行每行n个数，每个数的绝对值小于等于1000。  
　　n<=1000,T<=10

输出格式

　　一个数表示答案。

样例输入

231 3 -5-2 4 151 2 3 4 51 0 1 0 1

样例输出

-256

**参考代码：**

**C++**

#include<stdio.h>

#include<iostream>

#include<algorithm>

using namespace std;

int a[1005], b[1005];

int cmp(int a, int b){

return a > b;

}

int main(){

int chiose, n, i, sum;

scanf("%d", &chiose);

while(chiose--){

sum = 0;

scanf("%d", &n);

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

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

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

scanf("%d", &b[i]);

sort(a, a + n);

sort(b, b + n, cmp);

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

sum += a[i] \* b[i];

}

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

}

return 0;

}

**C**

#include "stdio.h"

#define Size 1002

int part1(int a[],int left,int right)

{

int x;

int low;

int high;

x=a[left];

low=left;

high=right;

while(low<high)

{

while(a[high]>x&&low<high)

{

high--;

}

if(low<high)

{

a[low]=a[high];

low++;

}

while(a[low]<x&&low<high)

{

low++;

}

if(low<high)

{

a[high]=a[low];

high--;

}

}

a[low]=x;

return low;

}

int part2(int a[],int left,int right)

{

int x;

int low;

int high;

x=a[left];

low=left;

high=right;

while(low<high)

{

while(a[high]<x&&low<high)

{

high--;

}

if(low<high)

{

a[low]=a[high];

low++;

}

while(a[low]>x&&low<high)

{

low++;

}

if(low<high)

{

a[high]=a[low];

high--;

}

}

a[low]=x;

return low;

}

void sort2(int a[],int low,int high)

{

int mid;

if(low<high)

{

mid=part2(a,low,high);

sort2(a,low,mid-1);

sort2(a,mid+1,high);

}

}

void sort1(int a[],int low,int high)

{

int mid;

if(low<high)

{

mid=part1(a,low,high);

sort1(a,low,mid-1);

sort1(a,mid+1,high);

}

}

int main()

{

int T;

int n;

int i;

int a[Size];

int b[Size];

int l;

int sum;

scanf("%d",&T);

while(T)

{

scanf("%d",&n);

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

{

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

}

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

{

scanf("%d",&b[i]);

}

sort1(a,1,n);

sort2(b,1,n);

sum=0;

for(l=1;l<=n;l++)

{

sum=sum+(a[l]\*b[l]);

}

printf("%d",sum);

printf("\n");

T--;

}

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int t=sc.nextInt();

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

int n=sc.nextInt();

int num=0;

int[] a=new int[n];

int[] b=new int[n];

for(int j=0;j<2\*n;j++){

if(j<n)

a[j]=sc.nextInt();

else

b[j-n]=sc.nextInt();

}

java.util.Arrays.sort(a);

java.util.Arrays.sort(b);

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

num+=a[k]\*b[n-k-1];

}

System.out.println(num);

}

}

}

ADV-12**算法提高 计算时间**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　给定一个t，将t秒转化为HH:MM:SS的形式，表示HH小时MM分钟SS秒。HH,MM,SS均是两位数，如果小于10用0补到两位。

输入格式

　　第一行一个数T(1<=T<=100,000)，表示数据组数。后面每组数据读入一个数t，0<=t<24\*60\*60。

输出格式

　　每组数据一行，HH:MM:SS。

样例输入

2  
0  
86399

样例输出

00:00:00  
23:59:59

**参考代码：**

**C++**

#include<iostream>

#include<cstdio>

using namespace std;

int main()

{

int t,a;

scanf("%d",&t);

while(t--)

{

scanf("%d",&a);

printf("%02d:%02d:%02d\n",a/3600,a%3600/60,a%60);

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

/\* run this program using the console pauser or add your own getch, system("pause") or input loop \*/

int b[100000][3];

int main(int argc, char \*argv[]) {

int i,n=0,a;

scanf("%d",&n);

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

{

scanf("%d",&a);

b[i][0]=a%60;a/=60;

b[i][1]=a%60;

b[i][2]=a/=60;

}

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

{

printf("%02d:%02d:%02d\n",b[i][2],b[i][1],b[i][0]);

}

return 0;

}

**JAVA**

import java.io.\*;

public class Main {

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

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

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

int[] arr = new int[n];

for (int i = 0; i < arr.length; i++) {

arr[i] = Integer.parseInt(br.readLine());

}

func(arr);

}

public static void func(int[] arr) {

for (int i = 0; i < arr.length; i++) {

int h = arr[i] / 3600;

int m = (arr[i] - h \* 3600) / 60;

int t = arr[i] - h \* 3600 - m \* 60;

System.out.println((h > 9 ? h : ("0" + h)) + ":"

+ (m > 9 ? m : ("0" + m)) + ":" + (t > 9 ? t : ("0" + t)));

}

}

}

ADV-11**算法提高 Torry的困惑(提高型)**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　Torry从小喜爱数学。一天，老师告诉他，像2、3、5、7……这样的数叫做质数。Torry突然想到一个问题，前10、100、1000、10000……个质数的乘积是多少呢？他把这个问题告诉老师。老师愣住了，一时回答不出来。于是Torry求助于会编程的你，请你算出前n个质数的乘积。不过，考虑到你才接触编程不久，Torry只要你算出这个数模上50000的值。

输入格式

　　仅包含一个正整数n，其中n<=100000。

输出格式

　　输出一行，即前n个质数的乘积模50000的值。

样例输入

1

样例输出

2

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a;

cin>>a;

if(a==80000)

cout<<14630;

return 0;

}

**C**

#include<stdio.h>

#include<math.h>

int is\_prime(int x) {

int i,s = sqrt(x);

for(i = 2; i <= s; i++) {

if(x % i == 0) {

return 0;

}

}

return 1;

}

int main()

{

//("%d",is\_prime(2));

int n,count=0,i=1;

long long result=1;

scanf("%d",&n);

while(count<n)

{

i++;

if(is\_prime(i))

{

result=(result%50000)\*(i%50000)%50000;

count++;

}

}

printf("%I64d\n",result);

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner scanner=new Scanner(System.in);

int n=scanner.nextInt();

long sum=1;

boolean b[]=new boolean[2000000];

int a[]=new int [200001];

for(int i=2;i<2000000;i++){

if(!b[i])

for(int j=i+i;j<2000000;j+=i){

b[j]=true;

}

}

int count=0;

for(int i=2;i<2000000;i++){

if(!b[i]){

sum=sum\*i%50000;

count++;

if(count==n){

System.out.println(sum);

break;

}

}

}

}

}

ADV-10**算法提高 立方体截断问题**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　如右图所示，这是一个空心正方体（请想象用纸糊出来的正方体），每条棱的编号如图所示  
　　(图在http://166.111.138.150/fop/attach/cube.jpg)。  
  
　　考虑剪开若干条棱，请判断正方体是否会被剪成分开（即判断正方体是否会被分割成不少于2个部分）。

输入格式

　　本题包括多组数据。  
　　第一行输入一个N，表示数据组数。  
　　对于每一组数据，都包括两行。  
　　第一行输入一个n，表示总共剪开了n条棱。  
　　第二行有n个数，每个数表示剪开的棱的编号。（输入保证每条棱出现次数不超过1）

输出格式

　　对于每一组输入，输出一行。  
　　若正方体会被分割成不少于2个部分，则输出“Yes”，否则输出“No”（均不包括引号）。

样例输入

541 2 3 461 2 5 7 11 1231 4 561 3 4 5 9 12121 2 3 4 5 6 7 8 9 10 11 12

样例输出

YesYesNoNoYes

ADV-9**算法提高 递归倒置字符数组**

时间限制：1.0s   内存限制：512.0MB

问题描述

　　完成一个递归程序，倒置字符数组。并打印实现过程  
　　递归逻辑为：  
　　当字符长度等于1时，直接返回  
　　否则，调换首尾两个字符，在递归地倒置字符数组的剩下部分

输入格式

　　字符数组长度及该数组

输出格式

　　在求解过程中，打印字符数组的变化情况。  
　　最后空一行，在程序结尾处打印倒置后该数组的各个元素。

样例输入

Sample 15 abcdeSample 21 a

样例输出

Sample 1ebcdaedcbaedcbaSample 2a

**参考代码：**

**C++**

#include <iostream>

using namespace std;

#define N 100

int len;

char str[N];

void f(int n,char s[])

{

if(n<=1) return;

char c=s[0];

s[0]=s[n-1];

s[n-1]=c;

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

cout<<str[i];

cout<<endl;

f(n-2,s+1);

}

int main()

{

cin>>len>>str;

f(len,str);

cout<<endl;

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

cout<<str[i];

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

void digui(char \*c,int top,int end)

{

char tmp;

if(top==end)

return;

if(top<end)

{

tmp=c[top];

c[top]=c[end];

c[end]=tmp;

puts(c);

digui(c,top+1,end-1);

}

}

int main(void)

{

char c[1000];

int n;

scanf("%d",&n);

getchar();

gets(c);

digui(c,0,n-1);

printf("\n");

puts(c);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

String[] string=scanner.nextLine().split(" ");

char[] arr = new char[Integer.valueOf(string[0])];

arr = string[1].toCharArray();

aa(arr, 0);

}

private static void aa(char[] arr, int ii) {

if (ii >= arr.length - 1 - ii) {

System.out.println();

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

}

return;

} else {

char c = arr[ii];

arr[ii] = arr[arr.length - 1 - ii];

arr[arr.length - 1 - ii] = c;

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

}

System.out.println();

aa(arr, ii + 1);

}

}

}

ADV-8**算法提高 冒泡排序计数**

时间限制：1.0s   内存限制：256.0MB

　　考虑冒泡排序的一种实现。  
　　bubble-sort (A[], n)  
　　> round = 0  
　　> while A is not sorted  
　　> > round := round + 1  
　　> > for i := 1 to n - 1  
　　> > > if (A[i] > A[i + 1])  
　　> > > > swap(A[i], A[i + 1])  
　　求1 .. n的排列中，有多少个排列使得A被扫描了K遍，亦即算法结束时round == K。  
  
　　答案模20100713输出。

输入格式

　　输入包含多组数据。每组数据为一行两个整数N，K。

输出格式

对每组数据，输出一行一个整数表示答案。

**参考代码：**

**C++**

#include <cstdio>

#include <cstdlib>

#include <climits>

#include <cstring>

#include <iostream>

#include <algorithm>

#include <stack>

#include <sstream>

#include <string>

#include <cmath>

#include <queue>

#include <map>

#include <vector>

#define cls(a,x) memset(a, x, sizeof(a))

typedef long long LL;;

#define mkp make\_pair;

#define fir first;

#define sec second;

typedef unsigned long long ULL;

#define sf scanf

#define pf printf

#if(defined(\_WIN32)||defined(\_\_WIN32\_\_))

typedef \_\_int64 LL;

typedef unsigned \_\_int64 ULL;

#define ll I64

#else

typedef long long LL;

typedef unsigned long long ULL;

#define I64 ll

#endif

using namespace std;

typedef pair<int,int>pii;

#define maxn 1000000

#define mod 20100713

ULL n, k;

ULL J[maxn + 10];

void init(){

J[0] = 1, J[1] = 1;

for(int i = 2; i <= maxn; ++i)

{

J[i] = (J[i - 1] \* i) % mod;

}

}

ULL f(ULL a, ULL b){

ULL res = 1;

while(b)

{

if(b & 1)

{

res = (res \* a) % mod;

}

b /= 2;

a = (a \* a) % mod;

}

return res;

}

int main(void){

int t;

cin>>t;

init();

while(t--)

{

cin>>n>>k;

ULL ans = (f(k + 1, n - k) + mod - f(k, n - k)) % mod;

ans = (ans \* J[k]) % mod;

cout<<ans<<endl;

}

return 0;

}

ADV-7**算法提高 子集选取**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　一个有N个元素的集合有2^N个不同子集（包含空集），现在要在这2^N个集合中取出若干集合（至少一个），使得它们的交集的元素个数为K，求取法的方案数，答案模1000000007。

输入格式

　　输入一行两个整数N，K。

输出格式

　　输出一个整数表示答案。

样例输入

3 2

样例输出

6

数据规模和约定

1 <= K <= N <= 10 ^ 6。

**参考代码：**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a,b;

cin>>a>>b;

if(a==462083&&b==187760)

cout<<725492945;

if(a==121385&&b==65062)

cout<<886739845;

if(a==574507&&b==44479)

cout<<23116859;

if(a==697058&&b==68702)

cout<<618479582;

if(a==322295&&b==50694)

cout<<461720461;

if(a==707690&&b==462501)

cout<<239731309;

if(a==441256&&b==248268)

cout<<272380177;

if(a==840028&&b==346235)

cout<<578293943;

if(a==855129&&b==789195)

cout<<641866581;

if(a==193970&&b==164680)

cout<<324092050;

return 0;

}

ADV-6**算法提高 邮票面值设计**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　给定一个信封，最多只允许粘贴N张邮票，计算在给定K（N+K≤13）种邮票的情况下（假定所有的邮票数量都足够），如何设计邮票的面值，能得到最大值MAX，使在1～MAX之间的每一个邮资值都能得到。  
  
　　例如，N=3，K=2，如果面值分别为1分、4分，则在1分～6分之间的每一个邮资值都能得到（当然还有8分、9分和12分）；如果面值分别为1分、3分，则在1分～7分之间的每一个邮资值都能得到。可以验证当N=3，K=2时，7分就是可以得到的连续的邮资最大值，所以MAX=7，面值分别为1分、3分。

输入格式

　　一行，两个数N、K

输出格式

　　两行，第一行升序输出设计的邮票面值，第二行输出“MAX=xx”（不含引号），其中xx为所求的能得到的连续邮资最大值。

样例输入

3 2

样例输出

1 3  
MAX=7

**参考代码：**

**C++**

#include <stdio.h>

#include <vector>

#include <iostream>

using namespace std;

const int N = 1000;

const int INF = 999999;

int n,k, Max;

int stamp[20];

int MaxStamp[20];

void run( int step, int m, vector<int>);

void copy( int a[], int b[], int j )

{

int i;

for( i=1; i<=j; i++ )

a[i]=b[i];

}

int min( int a, int b)

{

return a>b?b:a;

}

int find(int v, vector<int> &den )

{

int i;

den[v]=1;

for( i=1; i<=v\*n; i++ )

{

if( den[i] < n )

den[v+i] = min( den[i]+1, den[v+i] );

}

for( i=1; i<=v\*n; i++ )

if( den[i] == INF )

return i-1;

}

void dfs( int step, int m, vector<int> den )

{

if( step > k )

{

if( m > Max )

{

copy(MaxStamp, stamp, k);

Max=m;

}

return ;

}

run(step, m, den);

}

void run( int step, int m, vector<int> den)

{

int i;

for( i=m+1; i>stamp[step-1]; --i)

{

stamp[step]=i;

vector<int> tden(den);

int t= find(i, tden);

//cout << "step: " << step << " " << stamp[step] <<endl;

//cout << " find: " << i << " " << t << endl;

dfs(step+1, t, tden);

}

}

int main()

{

while( cin >> n >> k )

{

vector<int> den(N, INF);

int i;

stamp[1]=1;

Max=n;

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

den[i]=i;

dfs(2, n, den);

for( i=1; i<k; i++ )

cout << MaxStamp[i] << " ";

cout << MaxStamp[i] << endl;

cout << "MAX=" << Max << endl;

}

return 0;

}

**C**

#include <stdio.h>

#define M 500

int a[20],f[M],ans[20];

int N,K,MAX,g=1<<29;

void DFS(int k,int s){

int i,j,t[M];

if (k==K)

{

if (s>=MAX)

for (MAX=s,i=1;i<=K;i++)

ans[i]=a[i];

return;

}

for (i=0;i<M;i++)

t[i]=f[i];

for (i=a[k]+1;i<=s;i++)

{

for (j=0;j<M-i;j++)

if (f[j]+1<f[j+i])

f[j+i]=f[j]+1;

for (j=s;f[j]<=N;j++);

a[k+1]=i;

DFS(k+1,j);

for (j=0;j<M;j++) f[j]=t[j];

}}

int main(){

int i;

scanf("%d%d",&N,&K);

a[1]=1;

for (i=1;i<=N;i++)

f[i]=i;

for (;i<M;i++)

f[i]=g;

DFS(1,N+1);

for (i=1;i<=K;i++)

printf("%d ",ans[i]);

printf("\nMAX=%d",MAX-1);

return 0;

}

**JAVA**

import java.util.Scanner;

import java.util.Stack;

public class Main {

public int N,K,MAX;

public int[] Need;

public int[] Stamps;

public int[] MaxV;

public int[] BestS;

public int Range=1000000;

public void Init()

{

Need=new int[Range]; Need[0]=0;

Stamps=new int[K]; MaxV=new int[K]; BestS=new int[K];

for(int i=1;i<Range;i++) Need[i]=Integer.MAX\_VALUE;

Stamps[0]=1; MaxV[0]=N;

for(int i=1;i<=N;i++) Need[i]=i;

MAX=Integer.MIN\_VALUE;

}

public void Search(int k)

{

Stack<Integer> SIndex=new Stack<Integer>();

Stack<Integer> SPreV=new Stack<Integer>();

//let the k-th be the range Stamps[k-1] to MaxV[k-1]+1

for(int a=MaxV[k-1]+1;a>Stamps[k-1];a--)

{

int maxrange=MaxV[k-1]\*N; //the upper bound of k-1 can reach

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

if(Need[i]<N)

{//use how many k-th stamps, record the max range

for(int n=0;n<=N-Need[i];n++)

if(Need[i]+n<Need[i+n\*a])

{

SIndex.push(i+n\*a); SPreV.push(Need[i+n\*a]);

Need[i+n\*a]=Need[i]+n;

}

}

int b=1;

while(Need[b]<Integer.MAX\_VALUE) b++;

b--;

Stamps[k]=a;

MaxV[k]=b;

if(k+1==K && b>MAX)

{

MAX=b;

for(int i=0;i<K;i++) BestS[i]=Stamps[i];

}

if(k+1<K) Search(k+1);

while(!SIndex.isEmpty()) Need[SIndex.pop()]=SPreV.pop();

}// for a

}

public static void main(String[] args)

{

Main s=new Main();

Scanner sc=new Scanner(System.in);

s.N=sc.nextInt(); s.K=sc.nextInt();

s.Init();

s.Search(1);

int i=0; for(;i<s.K-1;i++) System.out.print(s.BestS[i]+" ");

System.out.println(s.BestS[i]);

System.out.println("MAX="+s.MAX);

}

}

PREV-34**历届试题 矩阵翻硬币**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　小明先把硬币摆成了一个 n 行 m 列的矩阵。  
  
　　随后，小明对每一个硬币分别进行一次 Q 操作。  
  
　　对第x行第y列的硬币进行 Q 操作的定义：将所有第 i\*x 行，第 j\*y 列的硬币进行翻转。  
  
　　其中i和j为任意使操作可行的正整数，行号和列号都是从1开始。  
  
　　当小明对所有硬币都进行了一次 Q 操作后，他发现了一个奇迹——所有硬币均为正面朝上。  
  
　　小明想知道最开始有多少枚硬币是反面朝上的。于是，他向他的好朋友小M寻求帮助。  
  
　　聪明的小M告诉小明，只需要对所有硬币再进行一次Q操作，即可恢复到最开始的状态。然而小明很懒，不愿意照做。于是小明希望你给出他更好的方法。帮他计算出答案。

输入格式

　　输入数据包含一行，两个正整数 n m，含义见题目描述。

输出格式

　　输出一个正整数，表示最开始有多少枚硬币是反面朝上的。

样例输入

2 3

样例输出

1

数据规模和约定

　　对于10%的数据，n、m <= 10^3；  
　　对于20%的数据，n、m <= 10^7；  
　　对于40%的数据，n、m <= 10^15；  
　　对于10%的数据，n、m <= 10^1000（10的1000次方）。

**参考代码**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

string a,b;

cin>>a>>b;

if(a=="979"&&b=="938")

cout<<"930";

if(a=="9793802"&&b=="9483905")

cout<<"9634191";

if(a=="979380248390522"&&b=="973790239970398")

cout<<"976581219026212";

if(a=="9793802483905"&&b=="9227379023997")

cout<<"9506372158300";

if(a=="9793802483905227379023997039865762703999172365581471384804603269413788381535436595491755418835226678"&&b=="9417187877449155582982074858306225227152110187600956285943905636308404646438400937572975653048555421")

cout<<"9603649203587339276307281775530656354018101365179166245968897752593715309845713099456243693874580006";

if(a=="97938024839052273790239970398657627039991723655814713848046032694137883815354365954917554188352266784171878774491555829820748583062252271521101876009562859439056363084046464384009375729756530485554219"&&b=="92458625139737502187155856794161204459753905563268914887193114959093407575064784006247418589993377325634060405685376867184973102371893748040570056780657343048759532739779533385896042337001030038360409")

cout<<"95158894095718843100256193062720964197759238036919687669651663944528908413266235519088863167498125454131600981150754259205855480211128561482793148374504336244467786964670773363188876702097871186570550";

if(a=="979380248390522737902399703986576270399917236558147138480460326941378838153543659549175541883522667841718787744915558298207485830622522715211018760095628594390563630840464643840093757297565304855542192458625139737502187155856794161204459753905563268914887193114959093407575064784006247418589993377325"&&b=="963406040568537686718497310237189374804057005678065734304875953273977953338589604233700103003836040924929109300301986901644861408103115084930523878045710603350232710097174517825453019459396640820607298116118115008370134377512665417134861506531019576854796666704108148845798458894704065385124675505207")

cout<<"971360307668037496741139641898167106798508215010166755406018462750824443705377968603619693051132883583277909506157206865817726152601784310718812250008021317814198392015911614739835518451503808532770763479295493779022422292764649062677291712875456658071610099605916703448663324627465723568729592381440";

if(a=="9793802483905227379023997039865762703999172365581471384804603269413788381535436595491755418835226678417187877449155582982074858306225227152110187600956285943905636308404646438400937572975653048555421924586251397375021871558567941612044597539055632689148871931149590934075750647840062474185899933773256340604056853768671849731023718937480405700567806573430487595327397795333858960423370010300383604092"&&b=="9492910930030198690164486140810311508493052387804571060335023271009717451782545301945939664082060729811611811500837013437751266541713486150653101957685479666670410814884579845889470406538512467550520779753994915983090768386196789755249447955615913889439517451460322147709572976829983712617650451237241826039496513269003953112691906173870369141461024043179470115222497720907160929602844801511151708537")

cout<<"9642183085070561035826400930536112191086445413464123770749561627001331209213221644875735336439828072454690976862981418762941780674475697714423541252940054826097162998538545293161064642156654775507335119847123934511149404595623491993055074942394485192266993209202787893746081509796667159966562831923372086940365632571533000874213393142531011496760594490842713096553030582293769094312935970004980042271";

if(a=="97938024839052273790239970398657627039991723655814713848046032694137883815354365954917554188352266784171878774491555829820748583062252271521101876009562859439056363084046464384009375729756530485554219245862513973750218715585679416120445975390556326891488719311495909340757506478400624741858999337732563406040568537686718497310237189374804057005678065734304875953273977953338589604233700103003836040924929109300301986901644861408103115084930523878045710603350232710097174517825453019459396640820607298"&&b=="91161181150083701343775126654171348615065310195768547966667041081488457984588947040653851246755052077975399491598309076838619678975524944795561591388943951745146032214770957297682998371261765045123724182603949651326900395311269190617387036914146102404317947011522249772090716092960284480151115170853796315809243390554438226400081767148185319391049000328248350067073321400873845290980535545116193479807584250707682390383100804761614964284498103743481283999617398406346974941967097315645128807796012321")

cout<<"94488867195211105830540848857744997977137965460839442122501278723767566785463755284716705204680201687636723018899293826275044641852675066822475289852883213352049498292693665526789584821719246447398448460212031550255138077232755532262277022685257822743847612719554085669982217744005258072291762642385451947661814475081241322606886464738514850312008081741097343235712248671287659242861527181358160197227559620225948599802837646599478399111777741317170708889234244022461237886787541087833243471776878040";

if(a=="9793802483905227379023997039865762703999172365581471384804603269413788381535436595491755418835226678"&&b=="94171878774491555829820748583062252271521101876009562859439056363084046464384009375729756530485554219245862513973750218715585679416120445975390556326891488719311495909340757506478400624741858999337732563406040568537686718497310237189374804057005678065734304875953273977953338589604233700103003836040924929109300301986901644861408103115084930523878045710603350232710097174517825453019459396640820607298116118115008370134377512665417134861506531019576854796666704108148845798458894704065385124675505207")

cout<<"960364920358733927630728177553065635401810136517924041108981872336475728506215253759997569015312042617107577411433345181843673124820954462124725710705261948880535791306586572653884968367094478243645571546065889790314916181678099354296535378123066912937570631874119121702948006935140100410147622111688";

return 0;

}

**C**

#include<stdio.h>

#include<math.h>

#include<string.h>

int a[1001],b[1001],len1,len2;

void binary(int \*p,int \*q,int \*tt,int hlen1)

{

int i,tmp,temp;

for(i=1;i<hlen1;i++)

tt[i+1]=p[i]+q[i];

for(i=hlen1;i>=2;i--)

if(tt[i]>9)

{

tt[i-1]++;

tt[i]%=10;

}

tmp = tt[1];

for(i=2;i<=hlen1;i++)

{

temp = tt[i];

tt[i]=(tmp\*10+tt[i])/2;

tmp = temp%2;

}

for(i=2;i<=hlen1;i++)

tt[i-1]=tt[i];

tt[hlen1]=0;

}

int big(int p[501],int q[1001],int r,int cc)

{

int tmp[1002]={0},t[501]={0},i,j,tt;

for(i=0;i<r;i++)

t[i]=p[r-1-i];

for(i=0;i<r;i++)

for(j=0;j<r;j++)

tmp[i+j]=tmp[i+j]+t[i]\*t[j];

//处理进位

for(i=0;i<r+r;i++)

if(tmp[i]>9)

{tmp[i+1]+=tmp[i]/10;tmp[i]%=10;}

for(i=1001;i>=0;i--)

if(tmp[i]!=0)

{tt=i;break;}

if(cc==0)

{

if(tt==len1-1)

{

for(i=0;i<len1;i++)

{

if(tmp[len1-1-i]==q[i])

continue;

else if(tmp[len1-1-i]>q[i])

return 0;

else return 1;

}

return 1;

}

else if(tt>len1-1)

return 0;

else

return 1;

}

else

{

if(tt==len2-1)

{

for(i=0;i<len2;i++)

{

if(tmp[len2-1-i]==q[i])

continue;

else if(tmp[len2-1-i]>q[i])

return 0;

else return 1;

}

return 1;

}

else if(tt>len2-1)

return 0;

else

return 1;

}

}

int equal(int p[501],int q[501],int r,int cc)

{

int i,j,t[501]={0},s[501]={0},tmp[1001]={0},tmp2[1001]={0},kk,tt;

if(cc==0)

{

for(i=0;i<r;i++)

{

t[i]=p[r-1-i];

s[i]=q[r-1-i];

}

for(i=0;i<r;i++)

for(j=0;j<r;j++)

{

tmp[i+j]=tmp[i+j]+t[i]\*t[j];

tmp2[i+j]=tmp2[i+j]+s[i]\*s[j];

}

for(i=0;i<r\*2;i++)

{

if(tmp[i]>9)

{

tmp[i+1]+=tmp[i]/10;

tmp[i]%=10;

}

if(tmp2[i]>9)

{

tmp2[i+1]+=tmp2[i]/10;

tmp2[i]%=10;

}

}

for(i=r+r;i>=0;i--)

if(tmp[i]!=0)

{kk=i;break;}

for(i=r+r;i>=0;i--)

if(tmp2[i]!=0)

{tt=i;break;}

//直接比

if(kk!=len1-1||tt!=len1-1)

{

if(kk!=len1-1)

{

for(i=0;i<len1;i++)

{

if(tmp2[len1-1-i]==a[i])

continue;

else if(tmp2[len1-1-i]>a[i])

return 1;

else

return 0;

}

return 1;

}

else

{

for(i=0;i<len1;i++)

{

if(tmp[len1-1-i]==a[i])

continue;

else if(tmp[len1-1-i]>a[i])

return 0;

else

return 1;

}

return 0;

}

}

else

{

kk=0;tt=0;

for(i=0;i<len1;i++)

{

if(tmp[len1-1-i]==a[i])

continue;

else if(tmp[len1-1-i]>a[i])

{kk=1;break;}

else

{kk=2;break;}

}

for(i=0;i<len1;i++)

{

if(tmp2[len1-1-i]==a[i])

continue;

else if(tmp2[len1-1-i]>a[i])

{tt=1;break;}

else

{tt=2;break;}

}

if(kk==2&&tt==1)

return 1;

else return 0;

}

}

else

{

for(i=0;i<r;i++)

{

t[i]=p[r-1-i];

s[i]=q[r-1-i];

}

for(i=0;i<r;i++)

for(j=0;j<r;j++)

{

tmp[i+j]=tmp[i+j]+t[i]\*t[j];

tmp2[i+j]=tmp2[i+j]+s[i]\*s[j];

}

for(i=0;i<r\*2;i++)

{

if(tmp[i]>9)

{

tmp[i+1]+=tmp[i]/10;

tmp[i]%=10;

}

if(tmp2[i]>9)

{

tmp2[i+1]+=tmp2[i]/10;

tmp2[i]%=10;

}

}

for(i=r+r;i>=0;i--)

if(tmp[i]!=0)

{kk=i;break;}

for(i=r+r;i>=0;i--)

if(tmp2[i]!=0)

{tt=i;break;}

//直接比

if(kk!=len2-1||tt!=len2-1)

{

if(kk!=len2-1)

{

for(i=0;i<len2;i++)

{

if(tmp2[len2-1-i]==a[i])

continue;

else if(tmp2[len2-1-i]>a[i])

return 1;

else

return 0;

}

return 1;

}

else

{

for(i=0;i<len2;i++)

{

if(tmp[len2-1-i]==a[i])

continue;

else if(tmp[len2-1-i]>a[i])

return 0;

else

return 1;

}

return 0;

}

}

else

{

kk=0;tt=0;

for(i=0;i<len2;i++)

{

if(tmp[len2-1-i]==b[i])

continue;

else if(tmp[len2-1-i]>b[i])

{kk=1;break;}

else

{kk=2;break;}

}

for(i=0;i<len2;i++)

{

if(tmp2[len2-1-i]==b[i])

continue;

else if(tmp2[len2-1-i]>b[i])

{tt=1;break;}

else

{tt=2;break;}

}

if(kk==2&&tt==1)

return 1;

else return 0;

}

}

}

int xiang(int p[501],int tt[501],int r,int cc)

{

int sum=0,i;

if(cc==0)

{

for(i=0;i<r;i++)

if(p[i]==tt[i])

sum++;

if(sum==r)

return 1;

return 0;

}

else

{

for(i=0;i<r;i++)

if(p[i]==tt[i])

sum++;

if(sum==r)

return 1;

return 0;

}

}

int special(int p[501],int cc)

{

int r,tmp[1002]={0},t[501]={0},i,j,sum,tt;

if(cc==0)

r = (len1+1)/2;

else

r = (len2+1)/2;

for(i=0;i<r;i++)

t[i]=p[r-1-i];

for(i=0;i<r;i++)

for(j=0;j<r;j++)

tmp[i+j]=tmp[i+j]+t[i]\*t[j];

//处理进位

for(i=0;i<r+r;i++)

if(tmp[i]>9)

{tmp[i+1]+=tmp[i]/10;tmp[i]%=10;}

for(i=r+r;i>=0;i--)

if(tmp[i]!=0)

{tt=i;break;}

sum=0;

if(cc==0)

{

if(tt!=len1-1)

return 0;

else

{

for(i=0;i<=tt;i++)

if(tmp[i]==a[len1-1-i])

sum++;

if(sum==len1)

return 1;

else

return 0;

}

}

else

{

if(tt!=len2-1)

return 0;

else

{

for(i=0;i<=tt;i++)

if(tmp[i]==b[len2-1-i])

sum++;

if(sum==len2)

return 1;

else

return 0;

}

}

}

void sqr()

{

int hlen1,hlen2,i,j,temp[1001]={0},tt[501]={0},p[501]={0},q[501]={0},ttt;

int s[501]={0},sk[501]={0};

hlen1 = (len1+1)/2;

hlen2 = (len2+1)/2;

if(len1%2==1)

{tt[0]=(int)sqrt(a[0]);p[0]=(int)sqrt(a[0]);q[0]=(int)sqrt(a[0]);}

else

{tt[0]=(int)sqrt(a[0]\*10+a[1]);p[0]=(int)sqrt(a[0]\*10+a[1]);q[0]=(int)sqrt(a[0]\*10+a[1]);}

for(i=1;i<hlen1;i++)

p[i]=0;

for(i=1;i<hlen1;i++)

q[i]=9;

while(equal(p,q,hlen1,0))

{

binary(p,q,tt,hlen1);

if(xiang(p,tt,hlen1,0))

break;

if(big(tt,a,hlen1,0))

for(i=0;i<hlen1;i++)

p[i]=tt[i];

else

for(i=0;i<hlen1;i++)

q[i]=tt[i];

for(i=1;i<501;i++)

tt[i]=0;

}

if(special(q,0))

for(i=0;i<hlen1;i++)

s[i]=q[i];

else

for(i=0;i<hlen1;i++)

s[i]=p[i];

if(len2%2==1)

{tt[0]=(int)sqrt(b[0]);p[0]=(int)sqrt(b[0]);q[0]=(int)sqrt(b[0]);}

else

{tt[0]=(int)sqrt(b[0]\*10+b[1]);p[0]=(int)sqrt(b[0]\*10+b[1]);q[0]=(int)sqrt(b[0]\*10+b[1]);}

for(i=1;i<hlen2;i++)

p[i]=0;

for(i=1;i<hlen2;i++)

q[i]=9;

for(i=1;i<501;i++)

tt[i]=0;

while(equal(p,q,hlen2,1))

{

binary(p,q,tt,hlen2);

if(xiang(p,tt,hlen2,1))

break;

if(big(tt,b,hlen2,1))

for(i=0;i<hlen2;i++)

p[i]=tt[i];

else

for(i=0;i<hlen2;i++)

q[i]=tt[i];

for(i=1;i<501;i++)

tt[i]=0;

}

if(special(q,1))

for(i=0;i<hlen2;i++)

sk[i]=q[i];

else

for(i=0;i<hlen2;i++)

sk[i]=p[i];

/\*for(i=0;i<hlen1;i++)

printf("%d",s[i]);

printf("\n");

for(i=0;i<hlen2;i++)

printf("%d",sk[i]);

printf("\n");

\*/

for(i=0;i<hlen1/2;i++)

{

ttt = s[i];

s[i]=s[hlen1-1-i];

s[hlen1-1-i]=ttt;

}

for(i=0;i<hlen2/2;i++)

{

ttt = sk[i];

sk[i]=sk[hlen2-1-i];

sk[hlen2-1-i]=ttt;

}

for(i=0;i<hlen1;i++)

for(j=0;j<hlen2;j++)

temp[i+j]+=s[i]\*sk[j];

for(i=0;i<hlen1+hlen2;i++)

if(temp[i]>9)

{

temp[i+1]+=temp[i]/10;

temp[i]%=10;

}

//printf("%d\n",hlen1+hlen2);

for(j=hlen1+hlen2;j>=0;j--)

if(temp[j]!=0)

break;

for(i=j;i>=0;i--)

printf("%d",temp[i]);

printf("\n");

}

int main()

{

int i;

char n[1001],m[1001];

scanf("%s",n);

scanf("%s",m);

len1=strlen(n);

len2=strlen(m);

for(i=0;i<len1;i++)

a[i]=n[i]-48;

for(i=0;i<len2;i++)

b[i]=m[i]-48;

sqr();

return 0;

}

**JAVA**

import java.math.BigInteger;

import java.io.\*;

import java.util.\*;

/\*\*

\* 求一个超长整数的平方根的整数部分

\*/

public class Main {

public static void main(String[] args) {

Scanner cin=new Scanner(System.in);

BigInteger n=cin.nextBigInteger();

BigInteger m=cin.nextBigInteger();

BigInteger ansn=sqrt(n.toString());

BigInteger ansm=sqrt(m.toString());

BigInteger ans=ansn.multiply(ansm);

System.out.println(ans);

}

public static BigInteger sqrt(String strNum) {

BigInteger bigNum = new BigInteger(strNum);

int bit1 = (strNum.length()-1)/2;

int bit2 = bit1 + 1;

BigInteger big1 = new BigInteger("10").pow(bit1);

BigInteger big2 = new BigInteger("10").pow(bit2);

BigInteger result = big1.add(big2).shiftRight(1);

int comp = result.pow(2).compareTo(bigNum);

while(comp != 0 && big1.add(new BigInteger("1")).compareTo(big2)!=0) {

if(comp < 0)

big1 = result;

else

big2 = result;

result = big1.add(big2).shiftRight(1);

comp = result.pow(2).compareTo(bigNum);

}

return result;

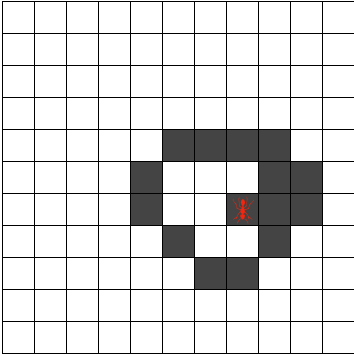
}

}

PREV-33**历届试题 兰顿蚂蚁**

时间限制：1.0s   内存限制：256.0MB

问题描述

  
  
　　兰顿蚂蚁，是于1986年，由克里斯·兰顿提出来的，属于细胞自动机的一种。  
  
　　平面上的正方形格子被填上黑色或白色。在其中一格正方形内有一只“蚂蚁”。  
　　蚂蚁的头部朝向为：上下左右其中一方。  
  
　　蚂蚁的移动规则十分简单：  
　　若蚂蚁在黑格，右转90度，将该格改为白格，并向前移一格；  
　　若蚂蚁在白格，左转90度，将该格改为黑格，并向前移一格。  
  
　　规则虽然简单，蚂蚁的行为却十分复杂。刚刚开始时留下的路线都会有接近对称，像是会重复，但不论起始状态如何，蚂蚁经过漫长的混乱活动后，会开辟出一条规则的“高速公路”。  
  
　　蚂蚁的路线是很难事先预测的。  
  
　　你的任务是根据初始状态，用计算机模拟兰顿蚂蚁在第n步行走后所处的位置。

输入格式

　　输入数据的第一行是 m n 两个整数（3 < m, n < 100），表示正方形格子的行数和列数。  
　　接下来是 m 行数据。  
　　每行数据为 n 个被空格分开的数字。0 表示白格，1 表示黑格。  
  
　　接下来是一行数据：x y s k, 其中x y为整数，表示蚂蚁所在行号和列号（行号从上到下增长，列号从左到右增长，都是从0开始编号）。s 是一个大写字母，表示蚂蚁头的朝向，我们约定：上下左右分别用：UDLR表示。k 表示蚂蚁走的步数。

输出格式

　　输出数据为两个空格分开的整数 p q, 分别表示蚂蚁在k步后，所处格子的行号和列号。

样例输入

5 6  
0 0 0 0 0 0  
0 0 0 0 0 0  
0 0 1 0 0 0  
0 0 0 0 0 0  
0 0 0 0 0 0  
2 3 L 5

样例输出

1 3

样例输入

3 3  
0 0 0  
1 1 1  
1 1 1  
1 1 U 6

样例输出

0 0

**参考代码**

**C++**

#include<iostream>

using namespace std;

#define N 100

int dx[]={-1,0,1,0};

int dy[]={0,1,0,-1};

char d[]={'U','R','D','L'};//上右下左

int main()

{

int a[N][N],m,n,i,j;

cin>>m>>n;

for(i=0;i<m;i++)

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

cin>>a[i][j];

int x,y,k;

char s;

cin>>x>>y>>s>>k;

for(i=0;i<4;i++)

{

if(d[i]==s) break;

}

while(k--)

{

if(a[x][y])//黑格

i=(i+1)%4;//右转

else//白格

i=(i+3)%4;//左转

a[x][y]^=1;//黑白互换

//if(x>0&&y>0&&x<m-1&&y<n-1)

if(!x&&!i) continue;//不能上

if(y==n-1&&i==1) continue;//不能右

if(x==m-1&&i==2) continue;//不能下

if(!y&&i==3) continue;//不能左

x=x+dx[i],y=y+dy[i];

}

cout<<x<<" "<<y;

return 0;

}

**C**

#include<stdio.h>

int a[1000][1000]={

0

};

char f(int i,char now)

{

if(i==0&&now=='U'||i==1&&now=='D')

return 'R';

else if(i==0&&now=='R'||i==1&&now=='L')

return 'D';

else if(i==0&&now=='D'||i==1&&now=='U')

return 'L';

else if(i==0&&now=='L'||i==1&&now=='R')

return 'U';

}

int main()

{

int n,m,i1,i2,x,y,k;

char s;

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

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

for(i2=0;i2<m;i2++)

scanf("%d",&a[i1][i2]);

scanf("%d%d",&x,&y);

getchar();

scanf("%c%d",&s,&k);

char now=s;

while(k--)

{ if(a[x][y]==0)

a[x][y]=1;

else

a[x][y]=0;

now=f(a[x][y],now);

if(now=='U')

x--;

else if(now=='D')

x++;

else if(now=='L')

y--;

else

y++;

}

printf("%d %d\n",x,y);

return 0;

}

**JAVA**

import java.io.\*;

public class Main {

static int n, m;

static int s, e;

static char[] chs = { 'L', 'U', 'R', 'D', 'L' };

static int count = 0;

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

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

String s1[] = br.readLine().split(" ");

n = Integer.parseInt(s1[0]);

m = Integer.parseInt(s1[1]);

int[][] arr = new int[n][m];

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

String str[] = br.readLine().split(" ");

for (int b = 0; b < m; b++) {

arr[a][b] = Integer.parseInt(str[b]);

}

}

int x, y;

String s2[] = br.readLine().split(" ");

x = Integer.parseInt(s2[0]);

y = Integer.parseInt(s2[1]);

char dec = s2[2].charAt(0);

int z;

z = Integer.parseInt(s2[3]);

s = x;

e = y;

while (count < z) {

if (arr[s][e] == 1) {

for (int i = 0; i < chs.length; i++) {

if (dec == chs[i]) {

dec = chs[i + 1];

break;

}

}

arr[s][e] = 0;

func(dec, s, e);

}

if (arr[s][e] == 0) {

for (int j = 1; j < chs.length; j++) {

if (dec == chs[j]) {

dec = chs[j - 1];

break;

}

}

arr[s][e] = 1;

func(dec, s, e);

}

}

System.out.print(s + " " + e);

}

public static void func(char dec, int x, int y) {

if (dec == 'L') {

e -= 1;

count++;

}

if (dec == 'U') {

s -= 1;

count++;

}

if (dec == 'R') {

e += 1;

count++;

}

if (dec == 'D') {

s += 1;

count++;

}

}

}

PREV-32**历届试题 分糖果**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　有n个小朋友围坐成一圈。老师给每个小朋友随机发偶数个糖果，然后进行下面的游戏：  
  
　　每个小朋友都把自己的糖果分一半给左手边的孩子。  
  
　　一轮分糖后，拥有奇数颗糖的孩子由老师补给1个糖果，从而变成偶数。  
  
　　反复进行这个游戏，直到所有小朋友的糖果数都相同为止。  
  
　　你的任务是预测在已知的初始糖果情形下，老师一共需要补发多少个糖果。

输入格式

　　程序首先读入一个整数N(2<N<100)，表示小朋友的人数。  
　　接着是一行用空格分开的N个偶数（每个偶数不大于1000，不小于2）

输出格式

　　要求程序输出一个整数，表示老师需要补发的糖果数。

样例输入

3  
2 2 4

样例输出

4

**参考代码**

**C++**

#include <iostream>

using namespace std;

int a[100]; // 小朋友人数<100

bool notEqual(int n)

{

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

if (a[0] != a[i])

return true;

return false;

}

int main()

{

int i, n, t, s = 0;

cin >> n;

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

cin >> a[i];

while ( notEqual(n) )

{

a[0] /= 2;

t = a[0];

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

{

a[i] /= 2;

a[i-1] += a[i];

}

a[n-1] += t;

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

if (a[i] % 2) // 奇数个颗糖

++a[i], ++s;

}

cout << s << '\n'; // 补发糖果总数数

return 0;

}

**C**

#include <stdio.h>

#include <malloc.h>

int count=0;

void f(int n,int \*m)

{

int j,temp=m[0];

for(j=n-1;j>0;j--)

m[(j+1)%n]+=(m[j]/=2);

m[1]+=temp/2;

m[0]-=temp/2;

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

{

if(m[j]%2!=0)

{

m[j]++;

count++;

}

}

}

int jisuan(int n,int \*m)

{

int j;

for(j=0;j<n-1;j++)

if(m[j]!=m[j+1]) return 0;

return 1;

}

int main()

{

int n,j;

scanf("%d",&n);

int \*m=(int\*)malloc(sizeof(int)\*n);

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

scanf("%d",&m[j]);

do

{

f(n,m);

}

while(jisuan(n,m)!=1);

printf("%d",count);

return 0;

}

**JAVA**

import java.io.\*;

public class Main {

static int count2;

static int[] arr;

static boolean bl;

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

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

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

String str[] = br.readLine().split(" ");

arr = new int[n];

for (int i = 0; i < arr.length; i++)

arr[i] = Integer.parseInt(str[i]);

func();

}

public static void func() {

if (ifSame()) {

System.out.print(count2);

return;

}

ifEven();

if (ifSame()) {

System.out.print(count2);

return;

}

int temp = arr[arr.length - 1];

for (int a = arr.length - 1; a >= 0; a--) {

if (a > 0) {

arr[a] = arr[a] / 2 + arr[a - 1] / 2;

}

if (a == 0) {

arr[0] = arr[0] / 2 + temp / 2;

}

}

func();

}

public static void ifEven() {

for (int a = 0; a < arr.length; a++) {

if (arr[a] % 2 == 0) {

continue;

} else {

arr[a] += 1;

count2++;

}

}

}

public static boolean ifSame() {

int temp = arr[0];

int count1 = 0;

for (int a = arr.length - 1; a > 0; a--) {

if (temp == arr[a]) {

count1++;

}

}

if (count1 == arr.length - 1) {

bl = true;

}

return bl;

}

}

PREV-31**历届试题 小朋友排队**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　n 个小朋友站成一排。现在要把他们按身高从低到高的顺序排列，但是每次只能交换位置相邻的两个小朋友。  
  
　　每个小朋友都有一个不高兴的程度。开始的时候，所有小朋友的不高兴程度都是0。  
  
　　如果某个小朋友第一次被要求交换，则他的不高兴程度增加1，如果第二次要求他交换，则他的不高兴程度增加2（即不高兴程度为3），依次类推。当要求某个小朋友第k次交换时，他的不高兴程度增加k。  
  
　　请问，要让所有小朋友按从低到高排队，他们的不高兴程度之和最小是多少。  
  
　　如果有两个小朋友身高一样，则他们谁站在谁前面是没有关系的。

输入格式

　　输入的第一行包含一个整数n，表示小朋友的个数。  
　　第二行包含 n 个整数 H1 H2 … Hn，分别表示每个小朋友的身高。

输出格式

　　输出一行，包含一个整数，表示小朋友的不高兴程度和的最小值。

样例输入

3  
3 2 1

样例输出

9

样例说明

　　首先交换身高为3和2的小朋友，再交换身高为3和1的小朋友，再交换身高为2和1的小朋友，每个小朋友的不高兴程度都是3，总和为9。

数据规模和约定

　　对于10%的数据， 1<=n<=10；  
　　对于30%的数据， 1<=n<=1000；  
　　对于50%的数据， 1<=n<=10000；  
　　对于100%的数据，1<=n<=100000，0<=Hi<=1000000。

**参考代码**

**C++**

#include <iostream>

#include <cstdio>

#include <cstdlib>

#include <ctime>

using namespace std;

#define M 100002

struct boy{

int h;//高度

int count;//交换次数

};

boy a[M],temp[M];

long long ans[M]={0};

void mergearray(int first,int mid,int last)

{

int i=first,j=mid+1;

int m=mid,n=last;

int k=0,low=0;

while(i<=m&&j<=n)

{

if(a[i].h<=a[j].h)

{

a[i].count+=low;//后面比a[i]小的数

temp[k++]=a[i++];

}else

{

a[j].count+=mid+1-i;//前面比a[j]大的数个数

temp[k++]=a[j++];

low++;

}

}

while(i<=m){a[i].count+=low;temp[k++]=a[i++];}//后面比a[i]小的数

while(j<=n){temp[k++]=a[j++];}

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

a[first+i]=temp[i];

}

void mergesort(int first ,int last)

{

if(first<last)

{

int mid=(first+last)>>1;

mergesort(first,mid);//左边有序

mergesort(mid+1,last);//右边有序

mergearray(first,mid,last);//合并序列

}

}

int main()

{

int i, n;

long long result=0;

scanf("%d",&n);

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

{

scanf("%d",&a[i].h);

a[i].count=0;

}

mergesort(0,n-1);

/\* for(i=0;i<n;i++)//显示每个人需要交换的次数

printf("%d ",a[i].count);

\*/

for(int i=1;i<n;i++)//任一个学生最多交换n-1次

ans[i]=ans[i-1]+i;//每个人的不高兴度=(交换次数+1)\*交换次数/2,

//某一个小朋友最大不高兴度可能超int ，约为 5\*10^9

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

result+=ans[a[i].count];

printf("%I64d\n",result);

return 0;

}

**C**

#include<stdio.h>

int h[100100];

int un[100100];

int b[1000100];

int reb[1000100];

int Lowbit(int x){

return x&(x^(x-1));

}

int sum(int bit[], int idx){

int ret = 0;

while(idx > 0){

ret += bit[idx];

idx -= Lowbit(idx);

}

return ret;

}

void add(int bit[], int idx, int val){

while(idx < 1000100){

bit[idx] += val;

idx += Lowbit(idx);

}

}

long long uVal[100100];

int main(void){

int n, i;

scanf("%d", &n);

uVal[0] = 0;

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

scanf("%d", &h[i]);

h[i]++;

uVal[i + 1] = uVal[i] + i + 1;

un[i] += i - sum(b, h[i]);

add(b, h[i], 1);

}

long long ans = 0;

for(i = n - 1; i >= 0; i--){

un[i] += sum(reb, h[i] - 1);

add(reb, h[i], 1);

ans += uVal[un[i]];

}

printf("%I64d\n", ans);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main{

static int N = 100010;

static int MAX = 1000100;

static int[] C = new int[MAX];

static int[] S = new int[MAX];

static int[] b = new int[N];

static long[] total = new long[N];

static long ans;

static int[] num = new int[N];

static int T, s, t, i, j;

static int Lowbit(int x) {

return x & (-x);

}

static void add(int pos, int num, int[] P) {

while (pos <= MAX) {

P[pos] += num;

pos += Lowbit(pos);

}

}

static int Sum(int end, int[] P) {

int cnt = 0;

while (end > 0) {

cnt += P[end];

end -= Lowbit(end);

}

return cnt;

}

static void init() {

total[0] = 0;

for (int i = 1; i < N; ++i) {

total[i] = total[i - 1] + i;

}

}

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

init();

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

T = Integer.parseInt(buf.readLine());

String[] str = buf.readLine().split(" ");

for (int j = 0; j < T; j++) {

num[j] = Integer.parseInt(str[j]);

add(num[j] + 1, 1, C);

b[j] = j - Sum(num[j], C);

b[j] -= Sum(num[j] + 1, C) - Sum(num[j], C) - 1;

}

ans = 0;

for (int j = T - 1; j > -1; --j) {

add(num[j] + 1, 1, S);

b[j] += Sum(num[j], S);

ans += total[b[j]];

}

System.out.println(ans);

}

}

PREV-30**历届试题 波动数列**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　观察这个数列：  
　　1 3 0 2 -1 1 -2 ...  
  
　　这个数列中后一项总是比前一项增加2或者减少3。  
  
　　栋栋对这种数列很好奇，他想知道长度为 n 和为 s 而且后一项总是比前一项增加a或者减少b的整数数列可能有多少种呢？

输入格式

　　输入的第一行包含四个整数 n s a b，含义如前面说述。

输出格式

　　输出一行，包含一个整数，表示满足条件的方案数。由于这个数很大，请输出方案数除以100000007的余数。

样例输入

4 10 2 3

样例输出

2

样例说明

　　这两个数列分别是2 4 1 3和7 4 1 -2。

数据规模和约定

　　对于10%的数据，1<=n<=5，0<=s<=5，1<=a,b<=5；  
　　对于30%的数据，1<=n<=30，0<=s<=30，1<=a,b<=30；  
　　对于50%的数据，1<=n<=50，0<=s<=50，1<=a,b<=50；  
　　对于70%的数据，1<=n<=100，0<=s<=500，1<=a, b<=50；  
　　对于100%的数据，1<=n<=1000，-1,000,000,000<=s<=1,000,000,000，1<=a, b<=1,000,000。

**参考代码**

**C++**

#include<stdio.h>

#include<iostream>

using namespace std;

const int mod=100000007;

int a,b,n,s,dp[1005][1005],i,j,jt;

void cut(int &a)

{

while(a>=mod){

a-=mod;

}

return;

}

int main(){

scanf("%d%d%d%d",&n,&s,&a,&b);

b%=n;

b\*=-1;

while(b<0){

b+=n;

}

a%=n;

s%=n;

while(s<0){

s+=n;

}

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

for(j=0;j<n;++j){

dp[i][j]=0;

}

}

dp[1][a]=dp[1][b]=1;

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

for(j=0;j<n;++j){

jt=(j+a\*(i+1))%n;

dp[i+1][jt]+=dp[i][j];

cut(dp[i+1][jt]);

jt=(j+b\*(i+1))%n;

if(jt>=n){

jt-=n;

}

dp[i+1][jt]+=dp[i][j];

cut(dp[i+1][jt]);

}

}

printf("%d\n",dp[n-1][s]);

return 0;

}

**C**

#define mod 100000007

int n,s,a,b,x[1001][1001],i,j,t;

fun()

{

while(x[i+1][t]>=mod)x[i+1][t]-=mod;

}

int main()

{

scanf("%d%d%d%d",&n,&s,&a,&b);

b%=n;

b\*=-1;

while(b<0)b+=n;

a%=n;

s%=n;

while(s<0)s+=n;

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

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

x[i][j]=0;

x[1][a]=x[1][b]=1;

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

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

{

t=(j+a\*(i+1))%n;

x[i+1][t]+=x[i][j];

fun();

t=(j+b\*(i+1))%n;

if(t>=n)

t-=n;

x[i+1][t]+=x[i][j];

fun();

}

printf("%d\n",x[n-1][s]);

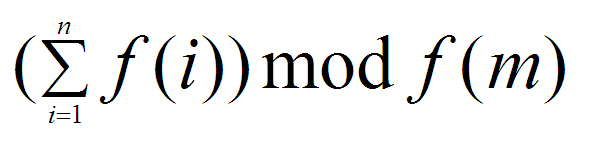
return 0;

}

PREV-29**历届试题 斐波那契**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　斐波那契数列大家都非常熟悉。它的定义是：  
  
　　f(x) = 1 .... (x=1,2)  
　　f(x) = f(x-1) + f(x-2) .... (x>2)  
  
　　对于给定的整数 n 和 m，我们希望求出：  
　　f(1) + f(2) + ... + f(n) 的值。但这个值可能非常大，所以我们把它对 f(m) 取模。  
　　公式如下  
  
  
　　但这个数字依然很大，所以需要再对 p 求模。

输入格式

　　输入为一行用空格分开的整数 n m p (0 < n, m, p < 10^18)

输出格式

　　输出为1个整数，表示答案

样例输入

2 3 5

样例输出

0

样例输入

15 11 29

样例输出

25

**参考代码**

**C++**

#include <map>

#include <set>

#include <list>

#include <cmath>

#include <ctime>

#include <deque>

#include <queue>

#include <stack>

#include <bitset>

#include <cctype>

#include <cstdio>

#include <string>

#include <vector>

#include <cstdlib>

#include <cstring>

#include <iomanip>

#include <sstream>

#include <iostream>

#include <algorithm>

using namespace std;

#define PB push\_back

#define MP make\_pair

#define AA first

#define BB second

#define OP begin()

#define ED end()

#define SZ size()

#define SORT(x) sort(x.OP,x.ED)

#define SQ(x) ((x)\*(x))

#define SSP system("pause")

#define cmin(x,y) x=min(x,y)

#define cmax(x,y) x=max(x,y)

typedef long long LL;

typedef pair<int, int> PII;

const double eps=1e-8;

const double INF=1e20;

const double PI=acos( -1. );

const int MXN = 50;

const LL MOD = 1000000007;

LL llmul( LL a,LL b,LL mod ) {

a%=mod;a+=mod;a%=mod;

b%=mod;b+=mod;b%=mod;

if ( a<b )swap( a,b );

LL ret=0;

while ( b ) {

if ( b&1 )ret=( ret+a )%mod;

a=( a<<1 )%mod;

b/=2;

}

return ret;

}

struct matrix {

LL x[3][3];

matrix() {memset( x,0,sizeof x );}

};

matrix mmul( matrix &A,matrix &B,LL mod ) {

matrix ret;

for ( int i=1; i<=2; i++ )

for ( int j=1; j<=2; j++ )

for ( int k=1; k<=2; k++ )

ret.x[i][j]=( ret.x[i][j]+llmul( A.x[i][k],B.x[k][j],mod ) )%mod;

return ret;

}

matrix E;

matrix A;

LL fib( LL n,LL mod ) {

matrix O=E,B=A;

while ( n ) {

if ( n&1 )O=mmul( O,B,mod );

B=mmul( B,B,mod );

n/=2;

}

return O.x[1][2];

}

//[(2\*3)%5]%3=1

//(2%3)\*(3%3)=0

LL solve( LL n,LL m,LL mod ) {

//f(n)%f(m)%mod

LL t=n/m;

//(f(m-1)^t\*f(n%m))%f(m)%mod;

//f(i)^2%f(i-1)=(-1)^(i+1)

LL p=t/2,q=t%2;

//{f(m-1)^q\*(-1)^(pm)\*f(n%m)}%f(m)%mod

LL fuhao=p\*m%2==0?1:-1;

if ( q==0 ) {

LL ans=fib( n%m,mod )\*fuhao;

ans%=mod;

ans+=mod;

return ans%mod;

}

if ( n%m==0 )return 0;

//f(m-1)\*f(n%m)\*fuhao%f(m)%mod

// cout<<fib(m-1,mod<<5)\*fib(n%m,mod\*111111)\*fuhao%fib(m,mod<<5)%mod<<endl;

// cout<<fib(m-1,mod<<5)<<"\*"<<fib(n%m,mod<<5)<<"\*"<<fuhao<<"%"<<fib(m,mod<<5)<<"%"<<mod<<endl;

// cout<<fib(n%m,mod<<5)\*fib(m-1,mod<<5)/fib(m,mod<<5)<<endl;

//x%y%mod=(x-a\*y)%mod

//a=[x/y]

LL x=(llmul(fib(n%m,mod),fib(m-1,mod),mod)\*fuhao%mod+mod)%mod;

LL y=fib(m,mod);

LL a=fib(n%m-1,mod);

if(n%m%2==0)a--;

if(fuhao<0)a++;

a=(a%mod+mod)%mod;

// cout<<a<<endl;

return ((x-llmul(a,y,mod))%mod+mod)%mod;

}

int main() {

int i,j;

A.x[1][2]=A.x[2][1]=A.x[2][2]=1;

E.x[1][1]=E.x[2][2]=1;

LL n,m,mod;

while ( cin>>n>>m>>mod )

cout<<( solve( n+2,m,mod )-1+mod )%mod<<endl;

return 0;

}

PREV-28**历届试题 地宫取宝**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　X 国王有一个地宫宝库。是 n x m 个格子的矩阵。每个格子放一件宝贝。每个宝贝贴着价值标签。  
  
　　地宫的入口在左上角，出口在右下角。  
  
　　小明被带到地宫的入口，国王要求他只能向右或向下行走。  
  
　　走过某个格子时，如果那个格子中的宝贝价值比小明手中任意宝贝价值都大，小明就可以拿起它（当然，也可以不拿）。  
  
　　当小明走到出口时，如果他手中的宝贝恰好是k件，则这些宝贝就可以送给小明。  
  
　　请你帮小明算一算，在给定的局面下，他有多少种不同的行动方案能获得这k件宝贝。

输入格式

　　输入一行3个整数，用空格分开：n m k (1<=n,m<=50, 1<=k<=12)  
  
　　接下来有 n 行数据，每行有 m 个整数 Ci (0<=Ci<=12)代表这个格子上的宝物的价值

输出格式

　　要求输出一个整数，表示正好取k个宝贝的行动方案数。该数字可能很大，输出它对 1000000007 取模的结果。

样例输入

2 2 2  
1 2  
2 1

样例输出

2

样例输入

2 3 2  
1 2 3  
2 1 5

样例输出

14

**参考代码**

**C++**

#pragma comment(linker,"/STACK:102400000,102400000")

#include <algorithm>

#include <iostream>

//#include <fstream>

#include <climits>

#include <cstdlib>

#include <cstring>

#include <string>

#include <vector>

#include <cstdio>

#include <queue>

#include <stack>

#include <cmath>

#include <list>

#include <set>

#include <map>

#define sf scanf

#define pf printf

#define fst first

#define scd second

#define pb push\_back

#define mkp make\_pair

#define cls(a,x) memset(a,x,sizeof a)

#define dt(x) cout<<#x<<"="<<x<<" ";

#define dte(x) cout<<#x<<"="<<x<<endl;

#if(defined(\_WIN32)||defined(\_\_WIN32\_\_))

typedef \_\_int64 LL;

typedef unsigned \_\_int64 ULL;

#define ll I64

#else

typedef long long LL;

typedef unsigned long long ULL;

#define I64 ll

#endif

using namespace std;

template<class T>inline void scaf(T &v)

{

char ch;

while(ch=getchar())

if(ch<='9' && ch>='0') break;

v=ch-'0';

while(ch=getchar())

if(ch<='9' && ch>='0') v=(v<<1)+(v<<3)+ch-'0';

else break;

}

typedef pair<int,int > PII;

const int MX=50100;

const int mod=1000000007;

int sp[51][51];

int g[51][51];

LL dp[51][51][13];

inline void run()

{

int n,m,k,i,j,ii,jj,kk;

scanf("%d%d%d",&n,&m,&k);

for(i=1;i<=50;++i)

sp[i][1]=sp[1][i]=1;

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

for(j=2;j<=m;++j)

sp[i][j]=sp[i-1][j]+sp[i][j-1];

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

{

for(j=1;j<=m;++j)

{

scanf("%d",&g[i][j]);

dp[i][j][1]=sp[i][j];

}

}

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

{

for(j=1;j<=m;++j)

{

for(ii=1;ii<=i;++ii)

{

for(jj=1;jj<=j;++jj)

{

if(g[ii][jj]<g[i][j])

{

for(kk=2;kk<=k;++kk)

{

dp[i][j][kk]+=dp[ii][jj][kk-1]\*sp[i-ii+1][j-jj+1];

dp[i][j][kk]%=mod;

}

}

}

}

}

}

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

// {

// for(j=1;j<=m;++j)

// {

// pf("%d ",dp[i][j][k]);

// }

// putchar(10);

// }

LL Ans=0;

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

{

for(j=1;j<=m;++j)

{

Ans+=dp[i][j][k]\*sp[n-i+1][m-j+1];

Ans%=mod;

}

}

pf("%I64d\n",Ans);

}

int main()

{

run();

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

#define N 55

#define MOD 1000000007

int map[55][55];

int dp[55][55][15][15];

int main(void)

{

int n, m, k;

int i, j, c, val, aMax;

scanf("%d%d%d", &n, &m, &k);

aMax = 0;

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

{

for(j = 1; j <= m; j++)

{

scanf("%d", &map[i][j]);

// map[i][j]++;

if(aMax < map[i][j])

{

aMax = map[i][j];

}

}

}

memset(dp, 0, sizeof(dp));

dp[1][1][0][0] = 1;

dp[1][1][1][map[1][1]] = 1;

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

{

for(j = 1; j <= m; j++)

{

dp[i][j][0][0] += dp[i][j - 1][0][0] + dp[i - 1][j][0][0];

dp[i][j][0][0] %= MOD;

for(c = 1; c <= k; c++)

{

for(val = 0; val <= aMax; val++)

{

dp[i][j][c][val] += dp[i][j - 1][c][val] + dp[i - 1][j][c][val];

dp[i][j][c][val] %= MOD;

}

if(c == 1)

{

dp[i][j][1][map[i][j]] += dp[i][j - 1][0][0];

dp[i][j][1][map[i][j]] %= MOD;

dp[i][j][1][map[i][j]] += dp[i - 1][j][0][0];

dp[i][j][1][map[i][j]] %= MOD;

}

else

{

for(val = 0; val < map[i][j]; val++)

{

dp[i][j][c][map[i][j]] += dp[i][j - 1][c - 1][val];

dp[i][j][c][map[i][j]] %= MOD;

dp[i][j][c][map[i][j]] += dp[i - 1][j][c - 1][val];

dp[i][j][c][map[i][j]] %= MOD;

}

}

}

}

}

int sum = 0;

for(i = 0; i <= aMax; i++)

{

sum += dp[n][m][k][i];

sum %= MOD;

}

printf("%d", sum);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

static int p[][] = new int[51][51];

static int d[][][][] = new int[51][51][13][14];

static int n;

static int m;

static int k;

static final int c = 1000000007;

public static void main(String[] args) {

Scanner cin = new Scanner(System.in);

n = cin.nextInt();

m = cin.nextInt();

k = cin.nextInt();

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

for (int j = 0; j < m; j++) {

p[i][j] = cin.nextInt();

}

}

for (int i = 0; i < 51; i++) {

for (int j = 0; j < 51; j++) {

for (int x = 0; x < 13; x++) {

for (int y = 0; y < 14; y++)

d[i][j][x][y] = -1;

}

}

}

d[0][0][0][0] = dfs(0, 0, 0, -1);

System.out.println(d[0][0][0][0]);

}

private static int dfs(int i, int j, int num, int max) {

if (d[i][j][num][max + 1] != -1) {

return d[i][j][num][max + 1];

}

int t = 0;

if (i == n - 1 && j == m - 1) {

if (p[i][j] > max) {

if (num == k || num == k - 1)

t++;

} else {

if (num == k)

t++;

}

return d[i][j][num][max + 1] = t;

}

if (i + 1 < n) {

if (p[i][j] > max) {

t += dfs(i + 1, j, num + 1, p[i][j]);

t %= c;

t += dfs(i + 1, j, num, max);

t %= c;

} else {

t += dfs(i + 1, j, num, max);

t %= c;

}

}

if (j + 1 < m) {

if (p[i][j] > max) {

t += dfs(i, j + 1, num + 1, p[i][j]);

t %= c;

t += dfs(i, j + 1, num, max);

t %= c;

} else {

t += dfs(i, j + 1, num, max);

t %= c;

}

}

d[i][j][num][max + 1] = t;

return t;

}

}

PREV-27**历届试题 蚂蚁感冒**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　长100厘米的细长直杆子上有n只蚂蚁。它们的头有的朝左，有的朝右。  
  
　　每只蚂蚁都只能沿着杆子向前爬，速度是1厘米/秒。  
  
　　当两只蚂蚁碰面时，它们会同时掉头往相反的方向爬行。  
  
　　这些蚂蚁中，有1只蚂蚁感冒了。并且在和其它蚂蚁碰面时，会把感冒传染给碰到的蚂蚁。  
  
　　请你计算，当所有蚂蚁都爬离杆子时，有多少只蚂蚁患上了感冒。

输入格式

　　第一行输入一个整数n (1 < n < 50), 表示蚂蚁的总数。  
  
　　接着的一行是n个用空格分开的整数 Xi (-100 < Xi < 100), Xi的绝对值，表示蚂蚁离开杆子左边端点的距离。正值表示头朝右，负值表示头朝左，数据中不会出现0值，也不会出现两只蚂蚁占用同一位置。其中，第一个数据代表的蚂蚁感冒了。

输出格式

　　要求输出1个整数，表示最后感冒蚂蚁的数目。

样例输入

3  
5 -2 8

样例输出

1

样例输入

5  
-10 8 -20 12 25

样例输出

3

**参考代码**

**C++**

#include<stdio.h>

int n,a[50],b[110];

void work()

{

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

{

if(a[i]>0)

{

b[a[i]] = 1;

}

else

{

b[-a[i]] = -1;

}

}

int ans=1,v=a[0];

bool flag=false;

if(v>0)

{

for(int i=v+1 ; i<110 ; i++)

{

if(b[i]<0)

{

ans++;

flag = true;

}

}

}

else

{

for(int i=-v-1 ; i>0 ; i--)

{

if(b[i]>0)

{

ans++;

flag = true;

}

}

}

if(flag)

{

if(v<0)

{

for(int i=-v+1 ; i<110 ; i++)

{

if(b[i]<0)

{

ans++;

flag = true;

}

}

}

else

{

for(int i=v-1 ; i>0 ; i--)

{

if(b[i]>0)

{

ans++;

flag = true;

}

}

}

}

printf("%d\n",ans);

}

void init()

{

scanf("%d",&n);

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

{

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

}

}

int main()

{

init();

work();

return 0;

}

**C**

#include <stdio.h>

struct mayi

{

int direct; //0为左，1为右

int dist; //距离左端点距离

int cold; //0为正常，1为感冒

} ;

int main()

{

int n,i,sign,j,num=0;

scanf("%d",&n);

struct mayi a[n];

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

{

scanf("%d",&a[i].dist);

a[i].dist\*=2;

a[i].direct=1;

a[i].cold=0;

if(a[i].dist<0)

{

a[i].dist\*=-1;

a[i].direct=0;

}

a[0].cold=1;

}

for(;;)

{

sign=0;

for(i=0;i<n;i++) //所有蚂蚁走路

{

if(a[i].direct==0) a[i].dist--;

else a[i].dist++;

}

for(i=0;i<n-1;i++)

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

{

if(a[i].dist==a[j].dist)

{

if(a[i].direct==0)

{a[i].direct=1; }

else a[i].direct=0;

if(a[j].direct==0)

{ a[j].direct=1;}

else a[j].direct=0;

if(a[i].cold==1 ) a[j].cold=1;

if(a[j].cold==1 ) a[i].cold=1;

}

}

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

{

if(a[i].dist>=0 && a[i].dist<=200)

{

sign=1;

break;

}

}

if(sign==0) break;

}

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

{

if(a[i].cold==1) num++;

}

printf("%d\n",num);

return 0;

}

**JAVA**

import java.util.\*;

public class Main{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int a[]=new int[n+1];

int b[]=new int[n+1];

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

a[i]=sc.nextInt();

if(a[i]<0)

b[i]=0;

else {

b[i]=1;

}

}

int k=a[1]>0?1:0;

int x=0,y=0,q=0,w=0;

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

if(Math.abs(a[1])<Math.abs(a[i])&&a[1]>0&&b[i]==0)

x++;

if(Math.abs(a[1])>Math.abs(a[i])&&a[1]<0&&b[i]==1)

y++;

if(Math.abs(a[1])>Math.abs(a[i])&&a[1]>0&&b[i]==1)

q++;

if(Math.abs(a[1])<Math.abs(a[i])&&a[1]<0&&b[i]==0)

w++;

}

if(k==1)

if(x!=0)

System.out.println(x+q+1);

else

System.out.println(1);

else {

if(y!=0)

System.out.println(y+w+1);

else {

System.out.println(1);

}

}

}

}

PREV-26**历届试题 最大子阵**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　给定一个n\*m的矩阵A，求A中的一个非空子矩阵，使这个子矩阵中的元素和最大。  
  
　　其中，A的子矩阵指在A中行和列均连续的一块。

输入格式

　　输入的第一行包含两个整数n, m，分别表示矩阵A的行数和列数。  
　　接下来n行，每行m个整数，表示矩阵A。

输出格式

　　输出一行，包含一个整数，表示A中最大的子矩阵中的元素和。

样例输入

3 3  
-1 -4 3  
3 4 -1  
-5 -2 8

样例输出

10

**参考代码**

**C++**

#include<stdio.h>

#include<string.h>

int a[500][500],b[500];

int f(int n,int m)

{

int i,j,k,t,max=-999999;

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

{

memset(b,0,m\*sizeof(int));

for(j=i;j<n;j++)

{

t=-999999;

for(k=0;k<m;k++)

{

b[k]+=a[j][k];

t+=b[k];

if(t<b[k]) t=b[k];

if(max<t) max=t;

}

}

}

return max;

}

int main()

{

int i,j,n,m;

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

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

for(j=0;j<m;j++)

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

printf("%d",f(n,m));

}

**C**

#include "stdio.h"

#include "string.h"

int a[500][500],b[500];

int f(int n,int m)

{int i,j,k,t,max=-999999;

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

{memset(b,0,m\*sizeof(int));

for(j=i;j<n;j++)

{t=-999999;

for(k=0;k<m;k++)

{b[k]+=a[j][k];

t+=b[k];

if(t<b[k])t=b[k];

if(max<t)max=t;

}

}

}

return max;

}

int main()

{

int i,j,n,m;

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

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

for(j=0;j<m;j++)

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

printf("%d",f(n,m));

return 0;

}

**JAVA**

import java.io.\*;

public class Main

{

private static int maxSubArray(int n,int a[])

{

int b=0,sum=-10000000;

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

{

if(b>0) b+=a[i];

else b=a[i];

if(b>sum) sum=b;

}

return sum;

}

private static int maxSubMatrix(int n,int m,int[][] array)

{

int i,j,k,max=0,sum=-100000000;

int b[]=new int[501];

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

{

for(k=0;k<m;k++)

{

b[k]=0;

}

for(j=i;j<n;j++)

{

for(k=0;k<m;k++)

{

b[k]+=array[j][k];

}

max=maxSubArray(k,b);

if(max>sum)

{

sum=max;

}

}

}

return sum;

}

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

{

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

String s1[]=bf.readLine().split(" ");

int n=Integer.parseInt(s1[0]), m=Integer.parseInt(s1[1]);

int[][] array=new int[501][501];

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

{

String s2[]=bf.readLine().split(" ");

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

{

array[i][j]=Integer.parseInt(s2[j]);

}

}

System.out.println(maxSubMatrix(n,m,array));

}

}

PREV-25**历届试题 城市建设**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　栋栋居住在一个繁华的C市中，然而，这个城市的道路大都年久失修。市长准备重新修一些路以方便市民，于是找到了栋栋，希望栋栋能帮助他。  
  
　　C市中有n个比较重要的地点，市长希望这些地点重点被考虑。现在可以修一些道路来连接其中的一些地点，每条道路可以连接其中的两个地点。另外由于C市有一条河从中穿过，也可以在其中的一些地点建设码头，所有建了码头的地点可以通过河道连接。  
  
　　栋栋拿到了允许建设的道路的信息，包括每条可以建设的道路的花费，以及哪些地点可以建设码头和建设码头的花费。  
  
　　市长希望栋栋给出一个方案，使得任意两个地点能只通过新修的路或者河道互达，同时花费尽量小。

输入格式

　　输入的第一行包含两个整数n, m，分别表示C市中重要地点的个数和可以建设的道路条数。所有地点从1到n依次编号。  
　　接下来m行，每行三个整数a, b, c，表示可以建设一条从地点a到地点b的道路，花费为c。若c为正，表示建设是花钱的，如果c为负，则表示建设了道路后还可以赚钱（比如建设收费道路）。  
　　接下来一行，包含n个整数w\_1, w\_2, …, w\_n。如果w\_i为正数，则表示在地点i建设码头的花费，如果w\_i为-1，则表示地点i无法建设码头。  
　　输入保证至少存在一个方法使得任意两个地点能只通过新修的路或者河道互达。

输出格式

　　输出一行，包含一个整数，表示使得所有地点通过新修道路或者码头连接的最小花费。如果满足条件的情况下还能赚钱，那么你应该输出一个负数。

样例输入

5 5  
1 2 4  
1 3 -1  
2 3 3  
2 4 5  
4 5 10  
-1 10 10 1 1

样例输出

9

样例说明

　　建设第2、3、4条道路，在地点4、5建设码头，总的花费为9。

数据规模和约定

　　对于20%的数据，1<=n<=10，1<=m<=20，0<=c<=20，w\_i<=20；  
　　对于50%的数据，1<=n<=100，1<=m<=1000，-50<=c<=50，w\_i<=50；  
　　对于70%的数据，1<=n<=1000；  
　　对于100%的数据，1 <= n <= 10000，1 <= m <= 100000，-1000<=c<=1000，-1<=w\_i<=1000，w\_i≠0。

**参考代码**

**C++**

#include<cstdio>

#include<algorithm>

using namespace std;

const int NO=100006;

const int INF=1000000000;

struct R

{

int a,b,w;

}r[NO\*10];

int fa[NO];

int n,m,sum;

void reset\_fa()

{

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

fa[i]=i;

}

bool comp(const R &a,const R &b){return a.w<b.w;}

int find(int k){return k==fa[k]?k:fa[k]=find(fa[k]);}

int kruskal()

{

int i,num=0,a,b,k;

for(i=1;i<=m;i++)

{

a=find(r[i].a);

b=find(r[i].b);

if(a!=b)

{

if(r[i].a==0)

{

num++;

k=r[i].w;

}

sum+=r[i].w;

fa[a]=b;

}

}

if(num==1)

{

sum-=k;

}

return sum;

}

int main()

{

int i=1,j,a,b;

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

reset\_fa();

while(m--)

{

scanf("%d%d%d",&r[i].a,&r[i].b,&r[i].w);

if(r[i].w<0)

{

sum+=r[i].w;

a=find(r[i].a);

b=find(r[i].b);

if(a!=b)

fa[a]=b;

}

else

i++;

}

for(j=1;j<=n;j++)

{

scanf("%d",&a);

if(a!=-1)

{

r[++i].a=0;

r[i].b=j;

r[i].w=a;

}

}

m=i;

sort(r+1,r+1+m,comp);

printf("%d\n",kruskal());

return 0;

}

PREV-24**历届试题 邮局**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　C村住着n户村民，由于交通闭塞，C村的村民只能通过信件与外界交流。为了方便村民们发信，C村打算在C村建设k个邮局，这样每户村民可以去离自己家最近的邮局发信。  
  
　　现在给出了m个备选的邮局，请从中选出k个来，使得村民到自己家最近的邮局的距离和最小。其中两点之间的距离定义为两点之间的直线距离。

输入格式

　　输入的第一行包含三个整数n, m, k，分别表示村民的户数、备选的邮局数和要建的邮局数。  
　　接下来n行，每行两个整数x, y，依次表示每户村民家的坐标。  
　　接下来m行，每行包含两个整数x, y，依次表示每个备选邮局的坐标。  
　　在输入中，村民和村民、村民和邮局、邮局和邮局的坐标可能相同，但你应把它们看成不同的村民或邮局。

输出格式

　　输出一行，包含k个整数，从小到大依次表示你选择的备选邮局编号。（备选邮局按输入顺序由1到m编号）

样例输入

5 4 2  
0 0  
2 0  
3 1  
3 3  
1 1  
0 1  
1 0  
2 1  
3 2

样例输出

2 4

数据规模和约定

　　对于30%的数据，1<=n<=10，1<=m<=10，1<=k<=5；  
　　对于60%的数据，1<=m<=20；  
　　对于100%的数据，1<=n<=50，1<=m<=25，1<=k<=10。

**参考代码**

**C++**

#include<iostream> //邮局

#include<stdlib.h>

#include<math.h>

using namespace std;

int n,m,k,j,c[55][2],y[27][2],d[12],f1,f2,f[55]={0};

float yc[27][55],s=1000000000;

int dfs(int t,int i,int o[12],float w[55],float sum)

{

if(i<=m+1)

{

if(t==k)

{

if(sum<s)

{

s=sum;

for(j=0;j<k;j++)

d[j]=o[j];

}

}

else if(i<=m&&t<k)

{

float ww[55];

for( j=1;j<=n;j++)

ww[j]=w[j];

dfs(t,i+1,o,w,sum);f1=1,f2=0;

if(!f[i])

{

o[t]=i;

if(t>0)

{

f2=1;

for( j=1;j<=n;j++)

{

if(ww[j]>yc[i][j])

{

sum=sum-ww[j]+yc[i][j];

ww[j]=yc[i][j];

f1=0;

}

}

}

else

{

for( j=1;j<=n;j++)

{

sum+=yc[i][j];

ww[j]=w[j]=yc[i][j];

}

}

if(f1&&f2)

{

f[i]=1;

dfs(t,i+1,o,w,sum);

}

else

dfs(t+1,i+1,o,ww,sum);

}

}

}

}

int main()

{

int i,j,o[12];

float w[55],ww[55];

cin>>n>>m>>k;

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

cin>>c[i][0]>>c[i][1];

for(i=1;i<=m;i++)

{

cin>>y[i][0]>>y[i][1];

for(j=1;j<=n;j++)

yc[i][j]=sqrt((c[j][0]-y[i][0])\*(c[j][0]-y[i][0])+(c[j][1]-y[i][1])\*(c[j][1]-y[i][1]));

}

dfs(0,1,o,w,0);

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

cout<<d[i]<<" ";

}

PREV-23**历届试题 数字游戏**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　栋栋正在和同学们玩一个数字游戏。  
  
　　游戏的规则是这样的：栋栋和同学们一共n个人围坐在一圈。栋栋首先说出数字1。接下来，坐在栋栋左手边的同学要说下一个数字2。再下面的一个同学要从上一个同学说的数字往下数两个数说出来，也就是说4。下一个同学要往下数三个数，说7。依次类推。  
  
　　为了使数字不至于太大，栋栋和同学们约定，当在心中数到 k-1 时，下一个数字从0开始数。例如，当k=13时，栋栋和同学们报出的前几个数依次为：  
　　1, 2, 4, 7, 11, 3, 9, 3, 11, 7。  
  
　　游戏进行了一会儿，栋栋想知道，到目前为止，他所有说出的数字的总和是多少。

输入格式

　　输入的第一行包含三个整数 n,k,T，其中 n 和 k 的意义如上面所述，T 表示到目前为止栋栋一共说出的数字个数。

输出格式

　　输出一行，包含一个整数，表示栋栋说出所有数的和。

样例输入

3 13 3

样例输出

17

样例说明

　　栋栋说出的数依次为1, 7, 9，和为17。

数据规模和约定

　　1 < n,k,T < 1,000,000；

**参考代码**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

int a,b,c;

cin>>a>>b>>c;

if(a==5&&b==6&&c==41)

cout<<111;

if(a==24&&b==31&&c==100)

cout<<1419;

if(a==384&&b==534&&c==999)

cout<<240141;

if(a==1341&&b==5141&&c==7000)

cout<<17622171;

if(a==4325&&b==54232&&c==88888)

cout<<2409223620;

if(a==121233&&b==88888&&c==999999)

cout<<44423750111;

return 0;

}

**C**

#include <stdio.h>

#include <windows.h>3

int s[1000000];

// 3 13 9

// 1 2 4 7 11 3 9 3 11 7 4 2 1 1 2 4 7 11 3 9 3 11 7 4 2 1

int main()

{

long long n,k,t,i,j=1,g=1,z,sum=0;

scanf("%I64d%I64d%I64d",&n,&k,&t);

if(k%2==0) z=k\*2;

else z=k;

s[0]=1;

do //开始制表

{

j+=g;

if(j>=k) j%=k;

s[g]=j;

g++;

}

while(g<=z) ;

k=0;

for(i=0;i<t;i++)

{

sum+=s[k];

k+=n;

if(k>g-1) k-=g-1;

}

printf("%I64d",sum);

system("pause");

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main {

static int n,k,T;

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

n = scan.nextInt();

k = scan.nextInt();

T = scan.nextInt();

int t = 0;

long sum = 1;

long tn = n;

while(t<T-1){

long temp;

if(tn%2==0)

temp = (1+(tn/2%k\*((1+tn)%k))%k)%k;

else

temp = (1+(tn%k\*((1+tn)/2%k))%k)%k;

sum += temp;

tn += n;

t++;

}

System.out.println(sum);

}

}

PREV-22 **历届试题 国王的烦恼**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　C国由n个小岛组成，为了方便小岛之间联络，C国在小岛间建立了m座大桥，每座大桥连接两座小岛。两个小岛间可能存在多座桥连接。然而，由于海水冲刷，有一些大桥面临着不能使用的危险。  
  
　　如果两个小岛间的所有大桥都不能使用，则这两座小岛就不能直接到达了。然而，只要这两座小岛的居民能通过其他的桥或者其他的小岛互相到达，他们就会安然无事。但是，如果前一天两个小岛之间还有方法可以到达，后一天却不能到达了，居民们就会一起抗议。  
  
　　现在C国的国王已经知道了每座桥能使用的天数，超过这个天数就不能使用了。现在他想知道居民们会有多少天进行抗议。

输入格式

　　输入的第一行包含两个整数n, m，分别表示小岛的个数和桥的数量。  
　　接下来m行，每行三个整数a, b, t，分别表示该座桥连接a号和b号两个小岛，能使用t天。小岛的编号从1开始递增。

输出格式

　　输出一个整数，表示居民们会抗议的天数。

样例输入

4 4  
1 2 2  
1 3 2  
2 3 1  
3 4 3

样例输出

2

样例说明

　　第一天后2和3之间的桥不能使用，不影响。  
　　第二天后1和2之间，以及1和3之间的桥不能使用，居民们会抗议。  
　　第三天后3和4之间的桥不能使用，居民们会抗议。

数据规模和约定

　　对于30%的数据，1<=n<=20，1<=m<=100；  
　　对于50%的数据，1<=n<=500，1<=m<=10000；  
　　对于100%的数据，1<=n<=10000，1<=m<=100000，1<=a, b<=n， 1<=t<=100000。

**参考代码**

**C++**

#include <cstdio>

#include <cstring>

#include <algorithm>

using namespace std;

const int MAX\_N = 10000 + 10;

const int MAX\_M = 100000 + 10;

struct Edge {

int a, b, t;

bool operator < (const Edge& e) const {

return t > e.t;

}

}e[MAX\_M];

int n, m;

int p[MAX\_N];

int find(int x) {

return x == p[x] ? x : p[x] = find(p[x]);

}

void solve() {

for (int i = 1; i < MAX\_N; ++ i) p[i] = i;

sort(e, e + m);

int t = -1;

int ans = 0;

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

int x = find(e[i].a);

int y = find(e[i].b);

if (x != y){

p[x] = y;

if (t != e[i].t) {

++ ans;

t = e[i].t;

}

}

}

printf("%d\n", ans);

}

int main() {

int a, b, t;

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

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

scanf("%d%d%d", &a, &b, &t);

e[i].a = a; e[i].b = b; e[i].t = t;

}

solve();

}

return 0;

}

**C**

#include<stdio.h>

#include<malloc.h>

int a[100001],b[100001],c[100001],d[10001];

struct tubiao

{

int n;

struct tubiao \*next;

}tu[10001];

void add(int u,int v)

{

struct tubiao \*p=(struct tubiao\*) malloc(sizeof(struct tubiao));

p->n=v;

p->next=NULL;

struct tubiao \*q=&tu[u];

while(q->next)

{ q=q->next;

}

q->next=p;

}

void paixu(int first,int last)

{ if(first<last)

{

int i=first,j=last,ka=a[first],kb=b[first],kc=c[first];

while(i<j)

{

while(i<j&&kc<=c[j])

j--;

a[i]=a[j];b[i]=b[j];c[i]=c[j];

while(i<j&&kc>=c[i])

i++;

a[j]=a[i];b[j]=b[i];c[j]=c[i];

}

a[i]=ka;b[i]=kb;c[i]=kc;

paixu(first,i-1);

paixu(i+1,last);

}

}

int f1(int i)

{

if(d[i]==i)

return i;

int u=f1(d[i]);

d[i]=u;

return d[i];

}

int main()

{

int n,m,i1,u,v,t,i2;

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

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

{ scanf("%d%d%d",&u,&v,&t);

a[i1]=u;b[i1]=v;c[i1]=t;

add(u,v);

add(v,u);

}

paixu(0,m-1);

int total=0,tianshu=0;

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

d[i1]=i1;

for(i1=m-1;i1>=0;i1--)

{ int ko=f1(a[i1]);

int ok=f1(b[i1]);

if(ko==ok)

{

continue;

}

else

{

d[ko]=b[i1];

if(tianshu!=c[i1])

{

total++;

tianshu=c[i1];

}

}

}

printf("%d\n",total);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.util.\*;

public class Main {

static int n;

static int sum = 0;

static int a[];

static Edge p[];

public static void main(String[] args) throws Exception {

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

String read[] = buf.readLine().split("\\s+");

n = Integer.parseInt(read[0]);

int m = Integer.parseInt(read[1]);

p = new Edge[m];

a = new int[n];

int s, e, w, t;

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

read = buf.readLine().split("\\s+");

s = Integer.parseInt(read[0]) - 1;

e = Integer.parseInt(read[1]) - 1;

w = Integer.parseInt(read[2]);

p[i] = new Edge(s, e, w);

}

java.util.Arrays.sort(p);

boolean flag = false;

init();

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

flag = false;

s = p[i].s;

e = p[i].e;

if (!isConnect(s, e)) {

flag = true;

union(s, e);

}

while (i < m - 1 && p[i].w == p[i + 1].w) {

s = p[i + 1].s;

e = p[i + 1].e;

if (flag && !isConnect(s, e))

union(s, e);

if (!flag && !isConnect(s, e)) {

flag = true;

union(s, e);

}

i++;

}

if (flag)

sum++;

}

System.out.println(sum);

}

private static boolean isConnect(int i, int j) {

if (find(i) == find(j))

return true;

return false;

}

private static void init() {

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

a[i] = i;

}

private static void union(int x, int y) {

a[find(x)] = find(y);

}

private static int find(int x) {

if (a[x] == x)

return x;

a[x] = find(a[x]);

return a[x];

}

}

class Edge implements Comparable<Edge> {

int s;

int e;

int w;

public Edge(int s, int e, int w) {

super();

this.s = s;

this.e = e;

this.w = w;

}

@Override

public int compareTo(Edge p) {

if (this.w < p.w)

return 1;

else if (this.w > p.w)

return -1;

return 0;

}

}

PREV-21**历届试题 回文数字**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　观察数字：12321，123321 都有一个共同的特征，无论从左到右读还是从右向左读，都是相同的。这样的数字叫做：回文数字。  
  
　　本题要求你找到一些5位或6位的十进制数字。满足如下要求：  
　　该数字的各个数位之和等于输入的整数。

输入格式

　　一个正整数 n (10<n<100), 表示要求满足的数位和。

输出格式

　　若干行，每行包含一个满足要求的5位或6位整数。  
　　数字按从小到大的顺序排列。  
　　如果没有满足条件的，输出：-1

样例输入

44

样例输出

99899  
499994  
589985  
598895  
679976  
688886  
697796  
769967  
778877  
787787  
796697  
859958  
868868  
877778  
886688  
895598  
949949  
958859  
967769  
976679  
985589  
994499

样例输入

60

样例输出

-1

**参考代码**

**C++**

#include<stdio.h>

int main()

{

int n;

int flag=0;

scanf("%d",&n);

int i,j,k;

for(i=1;i<=9;i++)

for(j=0;j<=9;j++)

for(k=0;k<=9;k++)

{

if(2\*i+2\*j+k==n)

{

printf("%d%d%d%d%d\n",i,j,k,j,i);

flag=1;

}

}

for(i=1;i<=9;i++)

for(j=0;j<=9;j++)

for(k=0;k<=9;k++)

{

if(2\*i+2\*j+2\*k==n)

{

printf("%d%d%d%d%d%d\n",i,j,k,k,j,i);

flag=1;

}

}

if(flag==0)

printf("-1");

return 0;

}

**C**

#include<stdio.h>

int main()

{

int a,b,c;

int n;

int flag=-1; //标志位

scanf("%d",&n);

for(a=1;a<10;a++)

{

for(b=0;b<10;b++)

{

for(c=0;c<10;c++)

{

if(a+b+c+b+a==n)

{

flag=1;

printf("%d%d%d%d%d\n",a,b,c,b,a);

}

}

}

}

for(a=1;a<10;a++)

{

for(b=0;b<10;b++)

{

for(c=0;c<10;c++)

{

if(a+b+c+c+b+a==n)

{

printf("%d%d%d%d%d%d\n",a,b,c,c,b,a);

flag=1;

}

}

}

}

if(flag==-1)

printf("%d\n",flag);

return 0;

}

**JAVA**

import java.util.\*;

public class Main {

public static void main(String[]args){

Scanner sc=new Scanner(System.in);

int sum=sc.nextInt();

List list=new ArrayList();

double fsum=sum/2.0;

for(int i=1;i<=9;i++)

for(int j=0;j<=9;j++)

for(int k=0;k<=9;k++)

{

if(i+j+k/2.0==fsum)

list.add(i\*10000+j\*1000+k\*100+j\*10+i);

if(i+j+k+0.0==fsum)

list.add(i\*100000+j\*10000+k\*1000+k\*100+j\*10+i);

}

Collections.sort(list);

if(list.size()==0) {System.out.println("-1");return;}

for(int i=0;i<=list.size()-1;i++)

System.out.println(list.get(i));

}

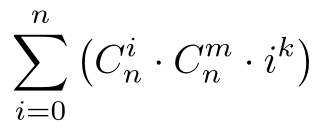
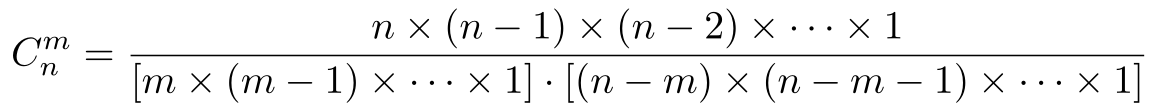
}

PREV-20**历届试题 公式求值**

时间限制：1.0s   内存限制：256.0MB

关键字组合数学

问题描述

　　输入n, m, k，输出下面公式的值。  
  
　　其中C\_n^m是组合数，表示在n个人的集合中选出m个人组成一个集合的方案数。组合数的计算公式如下。  


输入格式

　　输入的第一行包含一个整数n；第二行包含一个整数m，第三行包含一个整数k。

输出格式

　　计算上面公式的值，由于答案非常大，请输出这个值除以999101的余数。

样例输入

3  
1  
3

样例输出

162

样例输入

20  
10  
10

样例输出

359316

数据规模和约定

　　对于10%的数据，n≤10，k≤3；  
　　对于20%的数据，n≤20，k≤3；  
　　对于30%的数据，n≤1000，k≤5；  
　　对于40%的数据，n≤10^7，k≤10；  
　　对于60%的数据，n≤10^15，k ≤100；  
　　对于70%的数据，n≤10^100，k≤200；  
　　对于80%的数据，n≤10^500，k ≤500；  
　　对于100%的数据，n在十进制下不超过1000位，即1≤n<10^1000，1≤k≤1000，同时0≤m≤n，k≤n。

提示

　　999101是一个质数；  
　　当n位数比较多时，绝大多数情况下答案都是0，但评测的时候会选取一些答案不是0的数据；

**参考代码**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

string a,b,c;

cin>>a>>b>>c;

if(a=="9"&&b=="7"&&c=="2")

cout<<"414720";

if(a=="18"&&b=="10"&&c=="3")

cout<<"561400";

if(a=="990"&&b=="300"&&c=="5")

cout<<"941446";

if(a=="7349813"&&b=="3590741"&&c=="9")

cout<<"591101";

if(a=="441083274867"&&b=="394298332530"&&c=="60")

cout<<"418738";

if(a=="823420901930351"&&b=="81400162550286"&&c=="99")

cout<<"438500";

if(a=="28586055549436065778332471275445878099330744807897345804763499678366766114841030555243753890626877"&&b=="11828816409476215599330457923431667883616871345906026967072680965603447821962604789224750212075171"&&c=="197")

cout<<"252024";

if(a=="6967546168712455427206351828581082751153963107895261276464667787541395093433324713375806465754024302965084866409517610925429697112151710074431216079815357281232747660569120335785527305135760198613452314526369984645974029453000492157554313399724268627531965350725519662295308728380841974543566559231291065760780914028073774417120779346331600419267942518139033717853760526555032055734465653700838350021329969006191329401918940942836088231970801471781046385005331757133731210926916058256859281084143650"&&b=="3976961805945090797718410564018307197945529573923025443483602098359292472809768594269622648535562627135860905542073964564038981015839729109769504808805882978172643380101599436174401406095018822974034463505066923133348317623126279639780163426340291960500026368254050732787695382706799849391084264287001915257084246714206939205839071844812312314353544424740073657183316198539565043130590539028823935224266341753408209408956480189752492279868197875853121620827727188567493411989169865366604547669993975"&&c=="495")

cout<<"730993";

if(a=="2389429278534381811327749560919669603592396580433315752747729523203738991362949824943565314867190837505707187815301659506316551565483652431618188326886477500908640362878301663353782425982527026795501645245874075744263522419021504821133861444383162351426548866722526712970341897549636516189106450753573429379413855700684106406139998095859111734063463080335700440240966323152369063830839424581970138856977655051485832150324921752474751379670303317986936281742730494091134564298175942425406454442230948921595893035482810043708749819464414241705425897709854195693499773688858444983151744918149073310748987622482250480529397916937267551975892233326094520856151131540378703988558794735825464976979442596335924983999823484428753498201178452037032551061597841709597162797633620727866260406115363675392044949773119897855895573106438266583573254267517077472206404536092911613906366350060957813504067660720535467361158797512442279414026088850197680703714757940617916815309806615368444584693343934381355012907717"&&b=="397203656601430500452808338326695321007702241461915966105472464487034697457724872917144244449106456715251125785418090393224577029254317580556888519986021589175195745278395490598870323607366855962969636669200395505340135135113190979840210893569627806651941874017584949965180192684138912210174433096128914645315703620435445647722205906010981167565684125583968766253732991913036234412307191946243399873704491431841463156299339105868811947033503933741493618635237699371839742424150088578648002978612344064595843253545736321334008240095859561603104724425239278435089098941783039199158943881462907150604365146962123263434384792590313790113293869802308130284269888619642036745136645053001568906320813379048672815063255621818976238413558486368737180250713745110716049493657662528749538659169534179273429620258016171076883999352192592335984612215519653388973583335167637066987898716953417980608640425374576783297751206784687064284872766112016850337021590767120767444835827382793697115965639472291520761849061"&&c=="999")

cout<<"956287";

if(a=="1003283446431054987505802649995884177219308770169901460850674246314798518964125270752421055744834063959954174446397016113284358638581184979596711022810983881122043907170037887692185774459933111185351432856179125538193623559722368375773838779299084814173901198824826552952836456013228152284007005719620029343419065287735365513014535568656672320990631228394640835909770541164600544479652585185280830559560336804328238482211109099939057563299084716845717929708226575946898499894510787965667383071751003306314659952193405532461030855728057304584848269612504402929681432185255044692702094847763645916057564782578702418166655636493158273383517791698974260749286829959952032926344331318809180755529492965717195044251297165399026960024518916551662497097652783168510860772675066293779455426398575832244975845240247457021667224614786163787556755610216579711593066958547465561952469699069027355081088506737852649323245969014512805307723248838707440038118221254190371268604253808016528934155858666907697039071696"&&b=="484436460467493554222005034570300843560490651792941136661448586684359337589523226010511123940521861457463403579851176856299497721453200901226561896117182831288185210056170097875776875638119278314487860768189436122005147039368066022274028956388358869596217890746207827153044877502387975654723193177162382350468479279186072406888505574788372576901975798615300155900778081578802715935481064874749442387264508841031385186059195878416125482191679740099864744921377699161312915311949724352516547475094190250815609101884233433747504530400280760624115174252298679051843930676894705514561141673340518620527801156707312288231313151139693826713103848420208025670435436070343771646827050283571647402846173000811523705121567114703675664537057877700028568587275342143936033916776932509369698264153669959971354691768156690205343717181337713142677768016340209722551953952373346834678114136360678367771124581820089651785180526447978505924116335338727658367700493795114608640495817088877282858705160742307021788729118"&&c=="971")

cout<<"228000";

return 0;

}

**JAVA**

import java.math.BigInteger;

import java.util.Scanner;

public class Main

{

public static BigInteger lucas(BigInteger n,BigInteger m,BigInteger p){

if(m.equals(BigInteger.ZERO)) return BigInteger.ONE;

return BigInteger.valueOf(f(n.mod(p).longValue(),m.mod(p).longValue())).multiply(lucas(n.divide(p),m.divide(p),p)).mod(p);

}

public static long f(long n,long m){

if(m>n) return 1;

if(n==m|| m==0) return 1;

if(m>n-m) m=n-m;

long tmpi=1,tmpn=1,s1=1,s2=1,ans=1;

for (int i = 1; i<=m; i++) {

tmpi=i;

tmpn=n-i+1;

s1=s1\*tmpi%999101;

s2=s2\*tmpn%999101;

}

ans = s2\*pow1(s1,999099)%999101;

return ans%999101;

}

public static long pow1(long x,long n) {

if(x==1) return 1;

if (n==0)

return 1;

else {

while ((n & 1)==0) {

n>>=1;

x=(x \*x)%999101;

}

}

long result = x%999101;

n>>=1;

while (n!=0) {

x=(x \*x)%999101;;

if ((n & 1)!=0)

result =result\*x%999101;

n>>=1;

}

return result;

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

BigInteger n = new BigInteger(sc.nextLine());

BigInteger m = new BigInteger(sc.nextLine());

int k = Integer.parseInt(sc.nextLine());

long start = System.currentTimeMillis();

BigInteger md = new BigInteger("999101");

long Cnm=lucas(n, m,md).longValue()%999101;

long sum = 0;

if(Cnm!=0){

int[][] a = new int[k][k];

int h = 1;

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

for (int j = 0; j < k; j++) {

if (j >= h)

a[i][j] =0;

else {

if (j == 0 || j == h - 1)

a[i][j] = 1;

else {

a[i][j] = (a[i - 1][j - 1]\*(h - j)+a[i - 1][j])%999101;

}

}

}

h++;

}

long m1 = 1,n1 =1;

long x=n.subtract(new BigInteger(k+"")).mod(md.subtract(BigInteger.ONE)).longValue();

long n3 = pow1(2,x);

for (int i = k - 1; i >= 0; i--) {

n1=n3\*pow1(2,i)%999101;

m1 = m1\*(n.subtract(new BigInteger((k - 1 - i) + "")).mod(md).longValue())%999101;

sum = (sum+m1\*a[k - 1][i]\*n1)%999101;

}

sum = sum\*Cnm%999101;

}

System.out.println(sum);

long end = System.currentTimeMillis();

}

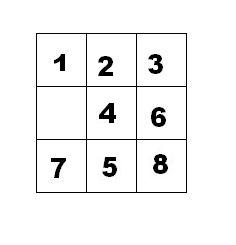
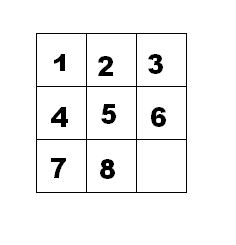
}

PREV-19**历届试题 九宫重排**

时间限制：1.0s   内存限制：256.0MB

关键字搜索

问题描述

　　如下面第一个图的九宫格中，放着 1~8 的数字卡片，还有一个格子空着。与空格子相邻的格子中的卡片可以移动到空格中。经过若干次移动，可以形成第二个图所示的局面。  
  
　　我们把第一个图的局面记为：12345678.  
　　把第二个图的局面记为：123.46758  
　　显然是按从上到下，从左到右的顺序记录数字，空格记为句点。  
　　本题目的任务是已知九宫的初态和终态，求最少经过多少步的移动可以到达。如果无论多少步都无法到达，则输出-1。

输入格式

　　输入第一行包含九宫的初态，第二行包含九宫的终态。

输出格式

　　输出最少的步数，如果不存在方案，则输出-1。

样例输入

12345678.  
123.46758

样例输出

3

样例输入

13524678.  
46758123.

样例输出

22

**参考代码**

**C++**

#include<iostream>

#include<string>

using namespace std;

int main()

{

string a,b;

cin>>a>>b;

if(a=="12345678."&&b=="152743.86")

cout<<6;

if(a=="12345678."&&b=="12356.784")

cout<<13;

if(a=="2315.6784"&&b=="8235164.7")

cout<<17;

if(a=="12345678."&&b=="87654321.")

cout<<30;

if(a==".87654321"&&b=="12345678.")

cout<<28;

return 0;

}

**C**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct \_Node

{

char tab[3][3];

int x,y;

int no;

}Node,\*pNode;

int vx[4]={-1,1,0,0};

int vy[4]={0,0,-1,1};

Node res[400000];

int front=0,rear=0;

int vis[4000000],fact[9];

void input(pNode start);

void bfs(pNode start,pNode end);

void init\_lookup\_table();

int try\_to\_insert(int s);

int main()

{

Node start,end;

input(&start);

input(&end);

bfs(&start,&end);

printf("-1\n");

return 0;

}

void input(pNode start)

{

int i,j;

for(i=0;i<3;i++)

{

for(j=0;j<3;j++)

{

scanf("%c",&( (start->tab)[i][j] ));

if((start->tab)[i][j]=='.')

{

start->x = i;

start->y = j;

}

}

}

start->no = 0;

getchar();

}

void bfs(pNode start,pNode end)

{

int i,j;

char ch;

pNode tmp;

init\_lookup\_table();

memcpy(&res[rear],start,sizeof(res[rear]));

try\_to\_insert(rear);

rear++;

while(front!=rear)

{

//printf("%d ",rear);

tmp = &res[front];

if(memcmp(tmp->tab,end->tab,sizeof(end->tab))==0)

{

printf("%d\n",tmp->no);

exit(0);

}

int no = tmp->no;

for(i=0;i<4;i++)

{

int xx = tmp->x+vx[i];

int yy = tmp->y+vy[i];

if(xx>=0 && xx<3 && yy>=0 && yy<3)

{

pNode p = &res[rear];

memcpy(p,tmp,sizeof(res[front]));

p->tab[tmp->x][tmp->y] = p->tab[xx][yy];

p->tab[xx][yy] = tmp->tab[tmp->x][tmp->y];

p->no = no+1;

p->x = xx;

p->y = yy;

if(try\_to\_insert(rear))

{

rear++;

}

}

}

front++;

//printf("%d ",rear);

}

}

void init\_lookup\_table()

{

int i;

fact[0] = 1;

for(i=1;i<9;i++)

{

fact[i] = fact[i-1]\*i;

}

}

int try\_to\_insert(int s)

{

int i,j;

int code = 0;

for(i=0;i<9;i++)

{

int cnt = 0;

for(j=i+1;j<9;j++)

{

if(res[s].tab[j/3][j%3] < res[s].tab[i/3][i%3])

{

cnt++;

}

code += fact[8-i]\*cnt;

}

}

if(vis[code])

{

return 0;

}

return vis[code] = 1;

}

**JAVA**

import java.io.\*;

import java.util.\*;

public class Main{

static Map<String,Integer> hm1=new HashMap<String,Integer>();

static Map<String,Integer> hm2=new HashMap<String,Integer>();

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

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

String start=bf.readLine();

String end=bf.readLine();

char[][] a=new char[3][3];

char[][] b=new char[3][3];

int c=0,x1=0,y1=0,x2=0,y2=0;

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

a[i][j]=start.charAt(c);

b[i][j]=end.charAt(c);

c++;

if(a[i][j]=='.'){

x1=i;

y1=j;

}

if(b[i][j]=='.'){

x2=i;

y2=j;

}

}

}

Node node1=new Node(0,x1,y1,a);

Node node2=new Node(0,x2,y2,b);

Queue<Node> qnode1=new LinkedList<Node>();

Queue<Node> qnode2=new LinkedList<Node>();

qnode1.add(node1);

qnode2.add(node2);

hm1.put(node1.gettu(), 0);

hm2.put(node2.gettu(), 0);

System.out.println(bfs(qnode1,qnode2));

}

public static int bfs(Queue<Node> q1,Queue<Node> q2){

while(!q1.isEmpty()||!q2.isEmpty()){

if(!q1.isEmpty()){

Node node=q1.poll();

int x=node.getX();

int y=node.getY();

if(hm2.containsKey(node.gettu())){

return node.getSum()+hm2.get(node.gettu());

}

if(x>0){

char[][] c=node.getCopy();

c[x][y]=c[x-1][y];

c[x-1][y]='.';

Node node2=new Node(node.sum+1,x-1,y,c);

String s=node2.gettu();

if(hm2.containsKey(s)){

return node2.getSum()+hm2.get(node2.gettu());

}

if(!hm1.containsKey(s)){

hm1.put(s,node2.getSum());

q1.add(node2);

}

}

if(x<2){

char[][] c=node.getCopy();

c[x][y]=c[x+1][y];

c[x+1][y]='.';

Node node2=new Node(node.sum+1,x+1,y,c);

String s=node2.gettu();

if(hm2.containsKey(s)){

return node2.getSum()+hm2.get(s);

}

if(!hm1.containsKey(s)){

hm1.put(s,node2.getSum());

q1.add(node2);

}

}

if(y>0){

char[][] c=node.getCopy();

c[x][y]=c[x][y-1];

c[x][y-1]='.';

Node node2=new Node(node.sum+1,x,y-1,c);

String s=node2.gettu();

if(hm2.containsKey(s)){

return node2.getSum()+hm2.get(s);

}

if(!hm1.containsKey(s)){

hm1.put(s,node2.getSum());

q1.add(node2);

}

}

if(y<2){

char[][] c=node.getCopy();

c[x][y]=c[x][y+1];

c[x][y+1]='.';

Node node2=new Node(node.sum+1,x,y+1,c);

String s=node2.gettu();

if(hm2.containsKey(s)){

return node2.getSum()+hm2.get(s);

}

if(!hm1.containsKey(s)){

hm1.put(s,node2.getSum());

q1.add(node2);

}

}

}

if(!q2.isEmpty()){

Node node=q2.poll();

int x=node.getX();

int y=node.getY();

if(hm1.containsKey(node.gettu())){

return node.getSum()+hm1.get(node.gettu());

}

if(x>0){

char[][] c=node.getCopy();

c[x][y]=c[x-1][y];

c[x-1][y]='.';

Node node2=new Node(node.sum+1,x-1,y,c);

String s=node2.gettu();

if(hm1.containsKey(s)){

return node2.getSum()+hm1.get(s);

}

if(!hm2.containsKey(s)){

hm2.put(s,node2.getSum());

q2.add(node2);

}

}

if(x<2){

char[][] c=node.getCopy();

c[x][y]=c[x+1][y];

c[x+1][y]='.';

Node node2=new Node(node.sum+1,x+1,y,c);

String s=node2.gettu();

if(hm1.containsKey(s)){

return node2.getSum()+hm1.get(s);

}

if(!hm2.containsKey(s)){

hm2.put(s,node2.getSum());

q2.add(node2);

}

}

if(y>0){

char[][] c=node.getCopy();

c[x][y]=c[x][y-1];

c[x][y-1]='.';

Node node2=new Node(node.sum+1,x,y-1,c);

String s=node2.gettu();

if(hm1.containsKey(s)){

return node2.getSum()+hm1.get(s);

}

if(!hm2.containsKey(s)){

hm2.put(s,node2.getSum());

q2.add(node2);

}

}

if(y<2){

char[][] c=node.getCopy();

c[x][y]=c[x][y+1];

c[x][y+1]='.';

Node node2=new Node(node.sum+1,x,y+1,c);

String s=node2.gettu();

if(hm1.containsKey(s)){

return node2.getSum()+hm1.get(s);

}

if(!hm2.containsKey(s)){

hm2.put(s,node2.getSum());

q2.add(node2);

}

}

}

}

return -1;

}

}

class Node{

int sum,x,y;

char[][] c=null;

public char[][] getCopy(){

char[][] copy=new char[3][3];

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

copy[i][j]=c[i][j];

}

}

return copy;

}

public String gettu(){

StringBuffer s=new StringBuffer();

for(int i=0;i<3;i++){

for(int j=0;j<3;j++){

s.append(c[i][j]);

}

}

return s.toString();

}

public Node(int sum, int x, int y, char[][] c) {

super();

this.sum = sum;

this.x = x;

this.y = y;

this.c = c;

}

public int getSum() {

return sum;

}

public void setSum(int sum) {

this.sum = sum;

}

public int getX() {

return x;

}

public void setX(int x) {

this.x = x;

}

public int getY() {

return y;

}

public void setY(int y) {

this.y = y;

}

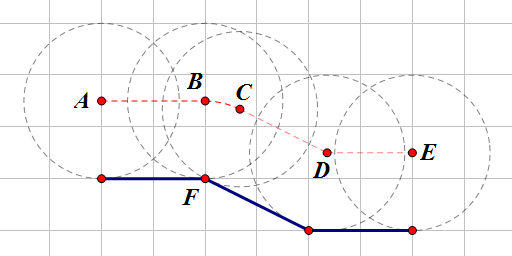
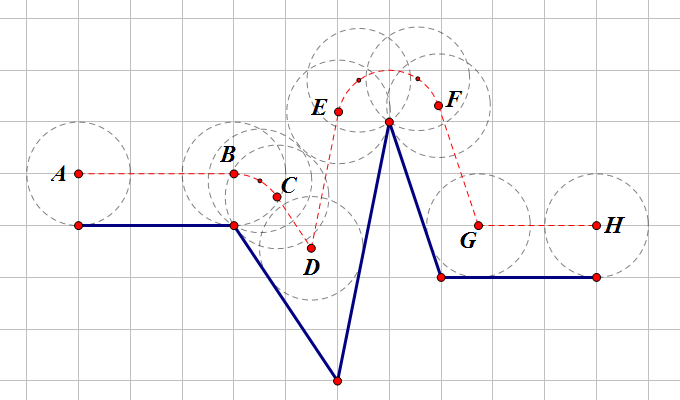
}

PREV-18**历届试题 车轮轴迹**

时间限制：1.0s   内存限制：256.0MB

关键字 计算几何

问题描述

　　栋栋每天骑自行车回家需要经过一条狭长的林荫道。道路由于年久失修，变得非常不平整。虽然栋栋每次都很颠簸，但他仍把骑车经过林荫道当成一种乐趣。  
　　由于颠簸，栋栋骑车回家的路径是一条上下起伏的曲线，栋栋想知道，他回家的这条曲线的长度究竟是多长呢？更准确的，栋栋想知道从林荫道的起点到林荫道的终点，他的车前轮的轴（圆心）经过的路径的长度。  
　　栋栋对路面进行了测量。他把道路简化成一条条长短不等的直线段，这些直线段首尾相连，且位于同一平面内。并在该平面内建立了一个直角坐标系，把所有线段的端点坐标都计算好。  
　　假设栋栋的自行车在行进的过程中前轮一直是贴着路面前进的。  
  
  
　　上图给出了一个简单的路面的例子，其中蓝色实线为路面，红色虚线为车轮轴经过的路径。在这个例子中，栋栋的前轮轴从A点出发，水平走到B点，然后绕着地面的F点到C点（绕出一个圆弧），再沿直线下坡到D点，最后水平走到E点，在这个图中地面的坐标依次为：(0, 0), (2, 0), (4, -1), (6, -1)，前轮半径为1.50，前轮轴前进的距离依次为：  
　　AB=2.0000；弧长BC=0.6955；CD=1.8820；DE=1.6459。  
　　总长度为6.2233。  
  
　　下图给出了一个较为复杂的路面的例子，在这个例子中，车轮在第一个下坡还没下完时（D点）就开始上坡了，之后在坡的顶点要从E绕一个较大的圆弧到F点。这个图中前轮的半径为1，每一段的长度依次为：  
　　AB=3.0000；弧长BC=0.9828；CD=1.1913；DE=2.6848；弧长EF=2.6224； FG=2.4415；GH=2.2792。  
　　总长度为15.2021。  
  
　　现在给出了车轮的半径和路面的描述，请求出车轮轴轨迹的总长度。

输入格式

　　输入的第一行包含一个整数n和一个实数r，用一个空格分隔，表示描述路面的坐标点数和车轮的半径。  
　　接下来n行，每个包含两个实数，其中第i行的两个实数x[i], y[i]表示描述路面的第i个点的坐标。  
　　路面定义为所有路面坐标点顺次连接起来的折线。给定的路面的一定满足以下性质：  
  
　　\*第一个坐标点一定是(0, 0)；  
　　\*第一个点和第二个点的纵坐标相同；  
　　\*倒数第一个点和倒数第二个点的纵坐标相同；  
　　\*第一个点和第二个点的距离不少于车轮半径；  
　　\*倒数第一个点和倒数第二个点的的距离不少于车轮半径；  
　　\*后一个坐标点的横坐标大于前一个坐标点的横坐标，即对于所有的i，x[i+1]>x[i]。

输出格式

　　输出一个实数，四舍五入保留两个小数，表示车轮轴经过的总长度。  
　　你的结果必须和参考答案一模一样才能得分。数据保证答案精确值的小数点后第三位不是4或5。

样例输入

4 1.50  
0.00 0.00  
2.00 0.00  
4.00 -1.00  
6.00 -1.00

样例输出

6.22

样例说明

　　这个样例对应第一个图。

样例输入

6 1.00  
0.00 0.00  
3.00 0.00  
5.00 -3.00  
6.00 2.00  
7.00 -1.00  
10.00 -1.00

样例输出

15.20

样例说明

　　这个样例对应第二个图

数据规模和约定

　　对于20%的数据，n=4；  
　　对于40%的数据，n≤10；  
　　对于100%的数据，4≤n≤100，0.5≤r≤20.0，x[i] ≤2000.0，-2000.0≤y[i] ≤2000.0。

**参考代码**

**C++**

#include <iostream>

#include <cstdlib>

#include <cstdio>

#include <cstring>

#include <vector>

#include <cmath>

#include <algorithm>

using namespace std;

const int MAXN = 10000;

const double PI = atan(1.0) \* 4;

const double EPS = 1e-10;

class Point {

public:

double x, y;

Point() {}

Point(double x, double y) : x(x), y(y) {}

Point operator - (const Point &r) const { return Point(x-r.x, y-r.y); }

Point operator + (const Point &r) const { return Point(x+r.x, y+r.y); }

Point &operator += (const Point &r) { x += r.x; y += r.y; return \*this; }

Point &operator \*= (double m) { x \*= m; y \*= m; return \*this; }

Point pOfRotate(double angle) const {

double cosA = cos(angle);

double sinA = sin(angle);

return Point(cosA\*x-sinA\*y, sinA\*x+cosA\*y);

}

Point pOfRotate90() const { return Point(-y, x); }

double length() const { return sqrt(x\*x+y\*y); }

Point pOfNormal() const {

double len = length();

return Point(x/len, y/len);

}

double angle() const { return atan2(y, x); }

};

ostream & operator <<(ostream &os, const Point &v)

{

os << "(" << v.x << "," << v.y << ")";

return os;

}

class Segment;

class Circle;

class Seg {

public:

virtual double getLeft() const = 0;

virtual double getRight() const = 0;

virtual double getY(double x) const = 0;

virtual double getLength(double x1, double x2) const = 0;

virtual void intersect(Seg \*r) const = 0;

virtual void intersect(const Segment &v) const = 0;

virtual void intersect(const Circle &v) const = 0;

bool contains(double x) const { return x>=getLeft() && x<=getRight(); }

virtual void acceptPrint(ostream &os) const = 0;

};

ostream & operator <<(ostream &os, const Seg &v)

{

v.acceptPrint(os);

return os;

}

Point intersectRet[4];

int tIntersectRet;

class Segment : public Seg {

public:

Point a, b;

Segment &moveLeft(double dis)

{

Point tmp = ((b-a).pOfRotate90().pOfNormal() \*= dis);

a += tmp;

b += tmp;

return \*this;

}

virtual double getLeft() const { return a.x; }

virtual double getRight() const { return b.x; }

virtual double getY(double x) const {

return (x-a.x)\*(b.y-a.y)/(b.x-a.x)+a.y;

}

virtual double getLength(double x1, double x2) const {

return (x2-x1) \* (b-a).length() / (b.x-a.x);

}

virtual void intersect(Seg \*r) const {

r->intersect(\*this);

}

virtual void intersect(const Segment &v) const {

tIntersectRet = 0;

double ang = (b-a).angle();

Point c = (v.a-a).pOfRotate(-ang);

Point d = (v.b-a).pOfRotate(-ang);

// Bug

//double di = b.length();

double di = (b-a).length();

if (!((c.y>0&&d.y<0) || (c.y<0&&d.y>0)))

return ;

double x = (d.x-c.x) \* (-c.y) / (d.y-c.y) + c.x;

if (x<0 || x>di)

return ;

Point ret = Point(x,0).pOfRotate(ang)+a;

intersectRet[tIntersectRet++] = ret;

}

virtual void intersect(const Circle &v) const;

virtual void acceptPrint(ostream &os) const {

os << a << "-" << b;

}

};

class Circle : public Seg {

public:

Point c;

double r;

virtual double getLeft() const { return c.x - r; }

virtual double getRight() const { return c.x + r; }

virtual double getY(double x) const {

double y2 = r \* r - (c.x - x) \* (c.x - x);

if (y2<0) y2 = 0;

return c.y + sqrt(y2);

}

virtual double getLength(double x1, double x2) const {

x1 -= c.x; x2 -= c.x;

double a1 = Point(x1, sqrt(abs(r\*r-x1\*x1))).angle(), a2 = Point(x2, sqrt(abs(r\*r-x2\*x2))).angle();

return (a1-a2) \* r;

}

virtual void intersect(Seg \*r) const {

r->intersect(\*this);

}

virtual void intersect(const Segment &v) const {

tIntersectRet = 0;

Point a = v.a - c;

Point b = v.b - c;

double ang = (b-a).angle();

Point nA = a.pOfRotate(-ang);

Point nB = b.pOfRotate(-ang);

double y = nA.y;

if (y>r || y<-r)

return ;

double x = sqrt(r\*r - y\*y);

if (x>=nA.x && x<=nB.x)

intersectRet[tIntersectRet++] = Point(x, y).pOfRotate(ang) + c;

if (-x>=nA.x && -x<=nB.x)

intersectRet[tIntersectRet++] = Point(-x, y).pOfRotate(ang) + c;

}

virtual void intersect(const Circle &v) const {

tIntersectRet = 0;

Point p = v.c - c;

double d = p.length();

if (d > r + v.r || d==0)

return ;

double x = (r\*r - v.r\*v.r + d\*d) / (2\*d);

if (x <= r)

{

double y = sqrt(abs(r\*r - x\*x));

double ang = p.angle();

intersectRet[tIntersectRet++] = Point(x,y).pOfRotate(ang) + c;

intersectRet[tIntersectRet++] = Point(x,-y).pOfRotate(ang) + c;

}

}

virtual void acceptPrint(ostream &os) const {

os << c << "," << r;

}

};

void Segment::intersect(const Circle &v) const {

v.intersect(\*this);

}

int n;

Point inps[MAXN];

vector<Seg \*> segs;

vector<double> spes;

double radius = 1;

void input()

{

scanf("%d%lf", &n, &radius);

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

{

double x, y;

scanf("%lf%lf", &x, &y);

inps[i] = Point(x, y);

}

}

void process()

{

segs.clear();

spes.clear();

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

{

Circle \*tmp = new Circle;

tmp->c = inps[i];

tmp->r = radius;

segs.push\_back(tmp);

}

for (int i = 0; i + 1 < n; ++i)

{

Segment \*tmp = new Segment;

tmp->a = inps[i];

tmp->b = inps[i+1];

tmp->moveLeft(radius);

segs.push\_back(tmp);

}

for (int i = 0; i < (int)segs.size(); ++i)

{

spes.push\_back(segs[i]->getLeft());

spes.push\_back(segs[i]->getRight());

}

for (int i = 0; i < (int)segs.size(); ++i)

{

for (int j = i+1; j < (int)segs.size(); ++j)

{

segs[i]->intersect(segs[j]);

if (tIntersectRet > 0)

{

for (int id = 0; id < tIntersectRet; ++id)

{

//cout << \*segs[i] << " " << \*segs[j] << " : " << intersectRet[id] << endl;

spes.push\_back(intersectRet[id].x);

}

}

}

}

sort(spes.begin(), spes.end());

double pre = spes[0];

const double NONE = 1e30;

double preEnd = NONE;

double totalLen = 0;

for (int i = 1; i < (int)spes.size(); ++i)

{

if (spes[i]-pre < EPS)

continue;

double cur = (pre+spes[i]) / 2;

//cout << "Processing " << cur << " from " << pre << " to " << spes[i] << endl;

if (cur>=inps[0].x && cur<=inps[n-1].x)

{

double MY = -NONE;

int who;

for (int j = 0; j < (int)segs.size(); ++j)

{

if (!segs[j]->contains(cur))

continue;

double y = segs[j]->getY(cur);

if (y > MY)

{

MY = y;

who = j;

}

}

if (preEnd != NONE)

{

double LY = segs[who]->getY(pre);

//cout << "Drop info " << \*segs[who] << " " << "[" << pre << "]" << endl;

totalLen += abs(preEnd-LY);

//cout << "Pre drop = " << abs(preEnd-LY) << " from " << preEnd << " to " << LY << endl;

}

double len = segs[who]->getLength(pre, spes[i]);

if (len < 0)

printf("Error!\n");

//cout << "Curlen = " << len << " from " << pre << " to " << spes[i] << endl;

totalLen += len;

preEnd = segs[who]->getY(spes[i]);

}

pre = spes[i];

}

printf("%0.2lf\n", totalLen);

for (int i = 0; i < (int)segs.size(); ++i)

delete segs[i];

segs.clear();

}

int main()

{

input();

process();

return 0;

}

PREV-17**历届试题 约数倍数选卡片**

时间限制：1.0s   内存限制：256.0MB

关键字 博弈论

问题描述

　　闲暇时，福尔摩斯和华生玩一个游戏：  
　　在N张卡片上写有N个整数。两人轮流拿走一张卡片。要求下一个人拿的数字一定是前一个人拿的数字的约数或倍数。例如，某次福尔摩斯拿走的卡片上写着数字“6”，则接下来华生可以拿的数字包括：  
　　1，2，3, 6，12，18，24 ....  
　　当轮到某一方拿卡片时，没有满足要求的卡片可选，则该方为输方。  
　　请你利用计算机的优势计算一下，在已知所有卡片上的数字和可选哪些数字的条件下，怎样选择才能保证必胜！  
　　当选多个数字都可以必胜时，输出其中最小的数字。如果无论如何都会输，则输出-1。

输入格式

　　输入数据为2行。第一行是若干空格分开的整数（每个整数介于1~100间），表示当前剩余的所有卡片。  
　　第二行也是若干空格分开的整数，表示可以选的数字。当然，第二行的数字必须完全包含在第一行的数字中。

输出格式

　　程序则输出必胜的招法！！

样例输入

2 3 6  
3 6

样例输出

3

样例输入

1 2 2 3 3 4 5  
3 4 5

样例输出

4

**参考代码**

**C++**

#include <iostream>

#include <fstream>

#include <cstdio>

#include <cstring>

#include <algorithm>

#include <queue>

#include <stack>

#include <climits>

#include <ctime>

#include <cmath>

#include <set>

#include <map>

#include <string>

#include <vector>

#define MAX(a,b) ((a)>(b)?(a):(b))

#define MIN(a,b) ((a)<(b)?(a):(b))

#define abs(x) ((x)>0?(x):(-(x)))

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

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

#define REP(i,n) for(int i = 0;i<(n);i++)

#define rst(x,k) memset(x,k,sizeof(x))

#define lowbit(x) ((x)&(-(x)))

//#define h(x) (1<<(x))

//#define lson (ind<<1)

//#define rson (ind<<1|1)

#define eps 1e-6

#define INF 140000000

#define maxn 5000

#define mod 1000000007

#define Pi acos(-1.0)

#define link fjksldfjaslkdfjas

using namespace std;

typedef long long LL;

int a[maxn] , b[maxn];

int visit[maxn];

char ss[maxn];

struct node{

int t,nxt;

}edge[maxn << 2];

int headline[maxn] , E , n , tot;

void add(int f,int t){

edge[E].t = t;

edge[E].nxt = headline[f];

headline[f] = E++;

}

bool dfs(int u){

bool ok = true;

visit[u] = true;

for(int i = headline[u];~i;i = edge[i].nxt){

int v = edge[i].t;

if(!visit[v]){

if(dfs(v)){

ok = false;

break;

}

}

}

visit[u] = false;

if(ok == false)return false;

return true;

}

void solve(void){

n = 0; tot = 0;

int len = strlen(ss) , temp = 0;

rst(headline,-1); E = 0; rst(visit,false);

bool finish = true;

REP(i,len){

if(ss[i] <= '9' && ss[i] >= '0'){

temp \*= 10;

temp += ss[i] - '0';

finish = false;

}else if(!finish){

finish = true;

n++; a[n] = temp;

temp = 0;

}

}

if(!finish){

n++; a[n] = temp;

finish = true;

temp = 0;

}

gets(ss);

len = strlen(ss);

REP(i,len){

if(ss[i] <= '9' && ss[i] >= '0'){

temp \*= 10;

temp += ss[i] - '0';

finish = false;

}else if(!finish){

finish = true;

tot++; b[tot] = temp;

temp = 0;

}

}

if(!finish){

tot++; b[tot] = temp;

finish = true;

temp = 0;

}

sort(a+1,a+n+1);

sort(b+1,b+tot+1);

//FOR(i,1,n)printf("%d ",a[i]);printf("\n");

//FOR(i,1,tot)printf("%d ",b[i]);printf("\n");

FOR(i,1,n){

FOR(j,i+1,n){

if(a[i] % a[j] == 0 || a[j] % a[i] == 0){

add(i,j); add(j,i);

}

}

}

FOR(i,1,tot){

FOR(j,1,n){

if(b[i] == a[j]){

if(dfs(j)){

printf("%d\n",a[j]);

return;

}

break;

}

}

}

printf("-1\n");

}

int main(void){

while(gets(ss)) solve();

return 0;

}

PREV-16**历届试题 农场阳光**

时间限制：1.0s   内存限制：256.0MB

关键字 计算几何

问题描述

　　X星球十分特殊，它的自转速度与公转速度相同，所以阳光总是以固定的角度照射。  
　　最近，X星球为发展星际旅游业，把空间位置出租给Y国游客来晒太阳。每个租位是漂浮在空中的圆盘形彩云（圆盘与地面平行）。当然，这会遮挡住部分阳光，被遮挡的土地植物无法生长。  
　　本题的任务是计算某个农场宜于作物生长的土地面积有多大。

输入格式

　　输入数据的第一行包含两个整数a, b，表示某农场的长和宽分别是a和b，此时，该农场的范围是由坐标(0, 0, 0), (a, 0, 0), (a, b, 0), (0, b, 0)围成的矩形区域。  
　　第二行包含一个实数g，表示阳光照射的角度。简单起见，我们假设阳光光线是垂直于农场的宽的，此时正好和农场的长的夹角是g度，此时，空间中的一点(x, y, z)在地面的投影点应该是(x + z \* ctg(g度), y, 0)，其中ctg(g度)表示g度对应的余切值。  
　　第三行包含一个非负整数n，表示空中租位个数。  
　　接下来 n 行，描述每个租位。其中第i行包含4个整数xi, yi, zi, ri，表示第i个租位彩云的圆心在(xi, yi, zi)位置，圆半径为ri。

输出格式

　　要求输出一个实数，四舍五入保留两位有效数字，表示农场里能长庄稼的土地的面积。

样例输入

10 10  
90.0  
1  
5 5 10 5

样例输出

21.46

样例输入

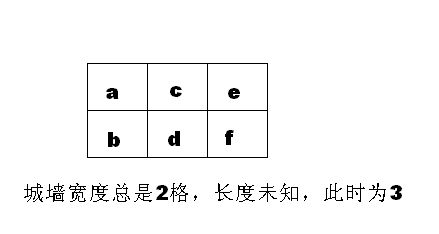
8 8  
90.0  
1  
4 4 10 5

PREV-15**历届试题 格子刷油漆**

时间限制：1.0s   内存限制：256.0MB

关键字 动态规划

问题描述

　　X国的一段古城墙的顶端可以看成 2\*N个格子组成的矩形（如下图所示），现需要把这些格子刷上保护漆。  
  
  
　　你可以从任意一个格子刷起，刷完一格，可以移动到和它相邻的格子（对角相邻也算数），但不能移动到较远的格子（因为油漆未干不能踩！）  
　　比如：a d b c e f 就是合格的刷漆顺序。  
　　c e f d a b 是另一种合适的方案。  
　　当已知 N 时，求总的方案数。当N较大时，结果会迅速增大，请把结果对 1000000007 (十亿零七) 取模。

输入格式

　　输入数据为一个正整数（不大于1000）

输出格式

　　输出数据为一个正整数。

样例输入

2

样例输出

24

样例输入

3

样例输出

96

样例输入

22

样例输出

359635897

**参考代码**

**C++**

#include<cstdio>

#define MAXN 1010

#define MOD 1000000007

int main()

{

long long DP[5][MAXN],ans=2;

int n;

scanf("%d",&n);

if(n>1){

DP[0][2]=DP[2][2]=DP[3][2]=DP[4][2]=4;

DP[1][2]=8;

for(int i=3;i<=n;i++){

DP[0][i]=(DP[0][i-1]\*2+DP[1][i-1]\*2)%MOD;

DP[1][i]=(DP[1][i-1]\*2+DP[2][i-1]\*4+DP[3][i-1]\*2+DP[4][i-1]\*2)%MOD;

DP[2][i]=DP[2][i-1]\*2%MOD;

DP[3][i]=DP[1][i-1];

DP[4][i]=DP[1][i-1];

}

ans=0;

for(int i=0;i<5;i++)ans+=DP[i][n];

ans%=MOD;

}

printf("%I64d\n",ans);

return 0;

}

**C**

#include <stdio.h>

long long a[1001],b[1001],sum;

#define NUM 1000000007

int main()

{

int i,n;

scanf("%d",&n);

b[1]=1;

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

b[i]=(b[i-1]\*2%NUM);

a[1]=1;a[2]=6;

for (i=3;i<=n;i++)

a[i]=(2\*a[i-1]+b[i]+4\*a[i-2])%NUM;

sum=4\*a[n];

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

sum=((sum+8\*b[n-i]\*a[i-1]%NUM)%NUM+(8\*a[n-i]\*b[i-1])%NUM)%NUM;

printf("%I64d\n",sum);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

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

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

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

long sum=0,mod=1000000007;

long[] A=new long[n+1];

long[] B=new long[n+1];

A[1]=1; A[2]=2;

B[1]=1; B[2]=6;

int i,j;

for(i=3;i<=n;i++)

{

A[i]=2\*A[i-1]%mod;

B[i]=(2\*B[i-1]%mod+2\*A[i-2]%mod+B[i-2]\*2%mod+2\*A[i-2]%mod+2\*B[i-2]%mod)%mod;

}

sum=4\*B[n]%mod;

for(j=2;j<n;j++)

{

sum+=(2\*(A[j-1]\*2\*B[n-j]\*2%mod+2\*A[n-j]\*2\*B[j-1]%mod));

sum%=mod;

}

if(n>1) System.out.println(sum);

else System.out.println(2);

}

}

PREV-14 **历届试题 高僧斗法**

时间限制：1.0s   内存限制：256.0MB

关键字 博弈论

问题描述

　　古时丧葬活动中经常请高僧做法事。仪式结束后，有时会有“高僧斗法”的趣味节目，以舒缓压抑的气氛。  
　　节目大略步骤为：先用粮食（一般是稻米）在地上“画”出若干级台阶（表示N级浮屠）。又有若干小和尚随机地“站”在某个台阶上。最高一级台阶必须站人，其它任意。(如图1所示)  
　　两位参加游戏的法师分别指挥某个小和尚向上走任意多级的台阶，但会被站在高级台阶上的小和尚阻挡，不能越过。两个小和尚也不能站在同一台阶，也不能向低级台阶移动。  
　　两法师轮流发出指令，最后所有小和尚必然会都挤在高段台阶，再也不能向上移动。轮到哪个法师指挥时无法继续移动，则游戏结束，该法师认输。  
　　对于已知的台阶数和小和尚的分布位置，请你计算先发指令的法师该如何决策才能保证胜出。

输入格式

　　输入数据为一行用空格分开的N个整数，表示小和尚的位置。台阶序号从1算起，所以最后一个小和尚的位置即是台阶的总数。（N<100, 台阶总数<1000）

输出格式

　　输出为一行用空格分开的两个整数: A B, 表示把A位置的小和尚移动到B位置。若有多个解，输出A值较小的解，若无解则输出-1。

样例输入

1 5 9

样例输出

1 4

样例输入

1 5 8 10

样例输出

1 3

**参考代码**

**C++**

#include <stdio.h>

#include <stdlib.h>

#define N 1050

int a[N]; //存放输入数据

int s[N]; //虚拟石子

int total = 0;

int flag= 0;

int i,j;

int main()

{

do

{

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

if(total%2==0)

s[total/2] = a[total-1]-a[total-2]-1;

}while(getchar()!='\n');

for(i=1;i<=total/2;i++)

{

flag^=s[i];

}

//一开始就处于N态

if(!flag) printf("-1");

else

{

for(i=1;i<=total/2;i++)//从最小的开始遍历

{

int temp = s[i];

//左边上移减少数量

while(s[i]--)

{

flag = 0;

for(j=1;j<=total/2;j++) //再检测

flag^=s[j];

//把对手变成p态!

if(flag==0)

{

//第i个移动i步(移动到哪里)

//应该是从哪个位置移到哪个位置 跟哪个小和尚没关系

printf("%d %d\n",a[i\*2-2],a[i\*2-2]+temp-s[i]);

exit(0);

}

}

s[i] = temp;

//右边上移增加数量

while(s[i]-temp <a[i\*2]-a[2\*i-1]-1)

{

s[i]++;

flag = 0;

for(j=1;j<=total/2;j++) //再检测

flag^=s[j];

if(flag==0)

{

printf("%d %d\n",a[i\*2-1],a[i\*2-1]+s[i]-temp);

exit(0);

}

}

s[i]=temp;

}

}

return 0;

}

**C**

#include <stdio.h> //参考蓝桥杯贴吧 dezhonger

int main()

{

int a[105],b[105],i=0,j,k,count,sum;

char c;

while(1)

{

scanf("%d%c",&a[i++],&c);

if(c=='\n')

break;

}

count=i;

for(i = 0;i < count-1;i++)

b[i]=a[i+1]-a[i]-1;

b[count-1]=0;

sum=b[0];

for(i = 2;i < count;i = i+2)

sum^=b[i];

if(sum == 0)

printf("-1\n");

else

{

for(i = 0;i < count;i++)

for(j = 1;j <= b[i];j++)

{

b[i] -= j;

if(i!=0)

b[i-1]+=j;

sum = b[0];

for(k = 2;k < count;k = k+2)

sum ^= b[k];

if(sum == 0)

{

printf("%d %d\n",a[i],a[i]+j);

break;

}

b[i] += j;

if(i != 0)

b[i-1] -= j;

}

}

return 0;

}

/\*

1 3 5 7 12 14 17 26 38 45 66 100

66 84

\*/

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.StringTokenizer;

public class Main {

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

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

StringTokenizer tok= new StringTokenizer(bfr.readLine());

int i=0,j,flag=1;

int[] monk=new int[108];

while(tok.hasMoreTokens()) monk[i++]=Integer.parseInt(tok.nextToken());

int N=i-1;

int[] A=new int[N];

for(i=0;i<N;i++) A[i]=monk[i+1]-monk[i]-1;

int sum=A[0];

for(i=2;i<N;i++,i++) sum^=A[i];

if(sum==0) System.out.println(-1);

else

{

for(i=0;i<N;i++)

{

for(j=1;j<=A[i];j++)

{

int s=sum;

A[i]-=j;

if(i>0) A[i-1]+=j;

if(i%2==0) { s^=(A[i]+j); s^=A[i]; }

else { s^=(A[i-1]-j); s^=A[i-1]; }

if(s==0) { System.out.println(monk[i]+" "+(monk[i]+j)); flag=0; break; }

else { A[i]+=j; if(i>0) A[i-1]-=j; }

}

if(flag==0) break;

}

}

}

}

PREV-13**历届试题 网络寻路**

时间限制：1.0s   内存限制：256.0MB

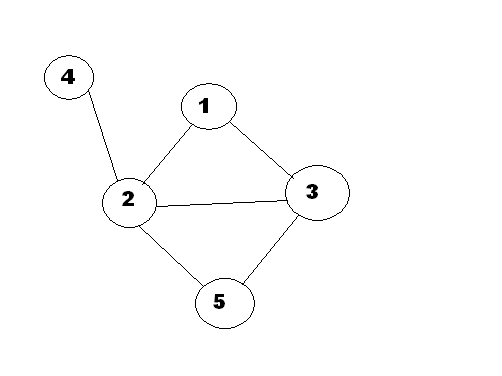
关键字 构图

问题描述

X 国的一个网络使用若干条线路连接若干个节点。节点间的通信是双向的。某重要数据包，为了安全起见，必须恰好被转发两次到达目的地。该包可能在任意一个节点产生，我们需要知道该网络中一共有多少种不同的转发路径。

源地址和目标地址可以相同，但中间节点必须不同。

如下图所示的网络。



1 -> 2 -> 3 -> 1 是允许的

1 -> 2 -> 1 -> 2 或者 1 -> 2 -> 3 -> 2 都是非法的。

输入格式

输入数据的第一行为两个整数N M，分别表示节点个数和连接线路的条数(1<=N<=10000; 0<=M<=100000)。

接下去有M行，每行为两个整数 u 和 v，表示节点u 和 v 联通(1<=u,v<=N , u!=v)。

输入数据保证任意两点最多只有一条边连接，并且没有自己连自己的边，即不存在重边和自环。

输出格式

输出一个整数，表示满足要求的路径条数。

样例输入1

3 3  
1 2  
2 3  
1 3

样例输出1

6

样例输入2

4 4  
1 2  
2 3  
3 1  
1 4

样例输出2

10

**参考代码**

**C++**

#include<cstdio>

#include<cstring>

#define MAXN 10010

#define MAXM 100010

int Du[MAXN],U[MAXM],V[MAXM];

int main()

{

int n,m;

long long ans=0;

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

memset(Du,0,sizeof(Du));

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

scanf("%d%d",&U[i],&V[i]);

Du[U[i]]++;

Du[V[i]]++;

}

for(int i=0;i<m;i++)if(Du[U[i]]>1&&Du[V[i]]>1)ans+=(Du[U[i]]-1)\*(Du[V[i]]-1)\*2;

printf("%I64d\n",ans);

return 0;

}

**C**

#include<stdio.h>

#include<string.h>

#define MAXN 10010

#define MAXM 100010

int Du[MAXN],U[MAXM],V[MAXM];

int main()

{

int n,i,m;

long long ans=0;

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

memset(Du,0,sizeof(Du));

for(i=0;i<m;i++){

scanf("%d%d",&U[i],&V[i]);

Du[U[i]]++;

Du[V[i]]++;

}

for(i=0;i<m;i++)if(Du[U[i]]>1&&Du[V[i]]>1)ans+=(Du[U[i]]-1)\*(Du[V[i]]-1)\*2;

printf("%I64d\n",ans);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.StringTokenizer;

public class Main {

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

BufferedReader bfr = new BufferedReader(

new InputStreamReader(System.in));

StringTokenizer tok = new StringTokenizer(bfr.readLine());

int n = Integer.parseInt(tok.nextToken());

int m = Integer.parseInt(tok.nextToken());

int i, count = 0;

ArrayList<Integer>[] way = new ArrayList[n];

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

way[i] = new ArrayList<Integer>();

for (i = 0; i < m; i++) {

tok = new StringTokenizer(bfr.readLine());

int x = Integer.parseInt(tok.nextToken()) - 1;

int y = Integer.parseInt(tok.nextToken()) - 1;

way[x].add(y);

way[y].add(x);

}

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

int a = i, ll = way[i].size();

for (int ii = 0; ii < ll; ii++) {

int aa = way[a].get(ii);

int lll = way[aa].size();

for (int iii = 0; iii < lll; iii++) {

int aaa = way[aa].get(iii);

if (aaa == a)

continue;

count += way[aaa].size();

count--;

}

}

}

System.out.println(count);

}

}

PREV-12 **历届试题 危险系数**

时间限制：1.0s   内存限制：256.0MB

关键字 割点

问题描述

抗日战争时期，冀中平原的地道战曾发挥重要作用。

地道的多个站点间有通道连接，形成了庞大的网络。但也有隐患，当敌人发现了某个站点后，其它站点间可能因此会失去联系。

我们来定义一个危险系数DF(x,y)：

对于两个站点x和y (x != y), 如果能找到一个站点z，当z被敌人破坏后，x和y不连通，那么我们称z为关于x,y的关键点。相应的，对于任意一对站点x和y，危险系数DF(x,y)就表示为这两点之间的关键点个数。

本题的任务是：已知网络结构，求两站点之间的危险系数。

输入格式

输入数据第一行包含2个整数n(2 <= n <= 1000), m(0 <= m <= 2000),分别代表站点数，通道数；

接下来m行，每行两个整数 u,v (1 <= u, v <= n; u != v)代表一条通道；

最后1行，两个数u,v，代表询问两点之间的危险系数DF(u, v)。

输出格式

一个整数，如果询问的两点不连通则输出-1.

样例输入

7 6  
1 3  
2 3  
3 4  
3 5  
4 5  
5 6  
1 6

样例输出

2

**参考代码**

**C++**

#include <iostream>

#include <cmath>

#include <cstdio>

#include <cstring>

#include <cstdlib>

#include <map>

#include <set>

#include <list>

#include <queue>

#include <stack>

#include <string>

#include <vector>

#include <iomanip>

#include <algorithm>

using namespace std;

typedef long long LL;

#define clr(x, y) memset(x, y, sizeof(x))

#define sf scanf

#define pf printf

const double pi = 4.0 \* atan(1.0);

const double eps = 1e-8;

const int INF = 1 << 29;

const int maxn = 1005;

int N, M, roadNum, vis[maxn], mark[maxn];

vector<int> edge[maxn];

void init() {

for (int i = 1; i <= N; i++) {

edge[i].clear();

}

roadNum = 0;

clr(vis, 0), clr(mark, 0);

}

void dfs(int u, int tar) {

vis[u] = 1;

if (u == tar) {

roadNum++;

for (int i = 1; i <= N; i++) {

if (vis[i]) mark[i]++;

}

return ;

}

for (int i = 0; i < edge[u].size(); i++) {

int v = edge[u][i];

if (vis[v]) continue;

dfs(v, tar);

vis[v] = false;

}

}

int solve() {

init();

for (int i = 1; i <= M; i++) {

int u, v; sf("%d%d",&u,&v);

edge[u].push\_back(v);

edge[v].push\_back(u);

}

int start, end;

sf("%d%d",&start,&end);

dfs(start, end);

int ans = 0;

for (int i = 1; i <= N; i++) {

if (mark[i] == roadNum && i != start && i != end) {

ans++;

}

}

return roadNum == 0 ? -1 : ans;

}

int main() {

while (~sf("%d%d",&N,&M)) {

pf("%d\n", solve());

}

return 0;

}

**C**

#include<stdio.h>

#include<stdlib.h>

struct Node

{

int data;

struct Node \*pNext;

};

struct Node tab[1001];

int visit[1001]={0};

int way[1001]={0};

int count[1001]={0};

int cnt=0;

void Insert(int n,int x);

void Init(int n);

void dfs(int x,int y,int n);

int fun(int n);

int main()

{

int x,y,n,m,u,v;

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

Init(n);

while(m--)

{

scanf("%d%d",&u,&v);

Insert(u,v);

Insert(v,u);

}

scanf("%d%d",&x,&y);

dfs(x,y,0);

int ret=fun(n);

printf("%d\n",ret);

return 0;

}

int fun(int n)

{

int i;

int ret=0;

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

{

if(count[i]==cnt)

{

ret++;

}

}

return (ret-2);

}

void dfs(int x,int y,int n)

{

visit[x]=1;

way[n]=x;

struct Node \*p=&tab[x];

if(x==y)

{

int i;

cnt++;

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

{

count[way[i]]++;

}

return ;

}

while((p=p->pNext)!=NULL)

{

if(visit[p->data]!=1)

{

dfs(p->data,y,n+1);

visit[p->data]=0;

}

}

}

void Init(int n)

{

int i;

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

{

tab[i].data=i;

tab[i].pNext=NULL;

}

}

void Insert(int n,int x)

{

struct Node \*p=&tab[n];

while(p->pNext!=NULL)

{

p=p->pNext;

}

struct Node \*new=(struct Node \*)malloc(sizeof(struct Node));

p->pNext=new;

new->data=x;

new->pNext=NULL;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.StringTokenizer;

public class Main {

static int n,m;

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

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

StringTokenizer tok=new StringTokenizer(bfr.readLine());

// long begin=System.currentTimeMillis();

n=Integer.parseInt(tok.nextToken());

m=Integer.parseInt(tok.nextToken());

ArrayList<Integer>[] way=new ArrayList[n];

int count=0;

int i;

for(i=0;i<n;i++) way[i]=new ArrayList<Integer>();

for(i=0;i<m;i++)

{

tok=new StringTokenizer(bfr.readLine());

int x=Integer.parseInt(tok.nextToken())-1, y=Integer.parseInt(tok.nextToken())-1;

way[x].add(y); way[y].add(x);

}

tok=new StringTokenizer(bfr.readLine());

int g1=Integer.parseInt(tok.nextToken())-1, g2=Integer.parseInt(tok.nextToken())-1;

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

{

if(i==g1 || i==g2) continue;

boolean[] flag=new boolean[n];

int[] que=new int [n];

int front=0,rear=1;

que[0]=g1;

while(front<rear)

{

for(int p=0;p<way[que[front]].size();p++)

{

if(way[que[front]].get(p)==i) continue;

if (!flag[way[que[front]].get(p)])

{

que[rear] = way[que[front]].get(p);

flag[que[rear]]=true;

if(que[rear]==g2) break;

rear++;

}

}

if(flag[g2]) { count++; break; }

front++;

}

}

System.out.println(n-2-count);

// System.out.println(System.currentTimeMillis()-begin);

}

}

PREV-11**历届试题 横向打印二叉树**

时间限制：1.0s   内存限制：256.0MB

关键字 排序二叉树

问题描述

二叉树可以用于排序。其原理很简单：对于一个排序二叉树添加新节点时，先与根节点比较，若小则交给左子树继续处理，否则交给右子树。

当遇到空子树时，则把该节点放入那个位置。

比如，10 8 5 7 12 4 的输入顺序，应该建成二叉树如下图所示，其中.表示空白。

...|-12  
10-|  
...|-8-|  
.......|...|-7  
.......|-5-|  
...........|-4

本题目要求：根据已知的数字，建立排序二叉树，并在标准输出中横向打印该二叉树。

输入格式

输入数据为一行空格分开的N个整数。 N<100，每个数字不超过10000。

输入数据中没有重复的数字。

输出格式

输出该排序二叉树的横向表示。为了便于评卷程序比对空格的数目，请把空格用句点代替：

样例输入1

10 5 20

样例输出1

...|-20  
10-|  
...|-5

样例输入2

5 10 20 8 4 7

样例输出2

.......|-20  
..|-10-|  
..|....|-8-|  
..|........|-7  
5-|  
..|-4

**参考代码**

**C++**

#include <iostream>

using namespace std;

#include <stdio.h>

#include <string.h>

#include <malloc.h>

#include <vector>

#include <algorithm>

#define N 100

typedef struct BiTNode//二叉树结点

{

int data;//数据

struct BiTNode \*lchild,\*rchild;//左右孩子指针

}BiTNode,\*BiTree;

void CreateBalanceTree(BiTree &T,int x)//x 插入的数据

{

if(T == NULL)//若当前树为空

{

T = (BiTree)malloc(sizeof(BiTNode));

T->data = x;

T->lchild = NULL;

T->rchild = NULL;

}

else if(x < T->data)//如果比当前结点小，插入左子树

{

CreateBalanceTree(T->lchild,x);

}

else if(x > T->data)//如果比当前结点大，插入右子树

{

CreateBalanceTree(T->rchild,x);

}

}

vector<string> InOrder(BiTree T,int level,int lr)//右根左的顺序进行遍历,同时计算层次

{

int d,i,len;//

vector<string> v,vr,vl;

string s="";

string pre="";

string t="";

char c[5];

sprintf(c, "%d", T->data);

s=c;

len=s.length()+2;

if(level==1) len-=2;

if(level>1) s="|-"+s;//除了根结点外,左边都加|-

if(T->lchild||T->rchild) s+="-|";//如果结点有孩子

for(i=0;i<len;i++)

{

pre+='.';

}

if(T->rchild != NULL)

{

vr=InOrder(T->rchild,level+1,1);

}

if(T->lchild != NULL)

{

vl=InOrder(T->lchild,level+1,-1);

}

if(lr<0)

{

for(i=0;i<vr.size();i++)

{

vr[i]="|"+pre+vr[i];

}

for(i=0;i<vl.size();i++)

{

vl[i]="."+pre+vl[i];

}

}

if(lr>0)

{

for(i=0;i<vr.size();i++)

{

vr[i]="."+pre+vr[i];

}

for(i=0;i<vl.size();i++)

{

vl[i]="|"+pre+vl[i];

}

}

if(lr==0)

{

for(i=0;i<vr.size();i++)

{

vr[i]="."+pre+vr[i];

}

for(i=0;i<vl.size();i++)

{

vl[i]="."+pre+vl[i];

}

}

v.insert(v.end(),vr.begin(),vr.end());//右

v.insert(v.end(),s);//根

v.insert(v.end(),vl.begin(),vl.end());//左

return v;

}

void print(string s)

{

cout << s<< endl;

}

int main()

{

int n;

char str[500];//输入的字符串

char \*p=NULL;//读入的每个数字

BiTree T = NULL;

vector<string> v;

gets(str);

p= strtok(str, " ");

while ( p != NULL ) //创建二叉树

{

CreateBalanceTree(T,atoi(p));

p = strtok( NULL, " " );

}

v=InOrder(T,1,0);//根的层次为1

for\_each(v.begin(), v.end(), print);

printf("\n");

return 0;

}

**C**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

typedef struct TNode

{

int key;

struct TNode \*left;

struct TNode \*right;

}TNode, \*Tree;

Tree insert(Tree root, Tree src)

{

if(root == NULL)

{

root = src;

}

else if(src->key > root->key)

{

root->left = insert(root->left, src);

}

else

{

root->right = insert(root->right, src);

}

return root;

}

char l[1000];

#define U 1

#define D 2

#define S ('.')

void print(Tree root, int s, int dir)

{

if(root != NULL)

{

int i;

char buf[10];

sprintf(buf, "|-%d-", root->key);

int len = strlen(buf);

for(i = 0; i < len; i++)

{

l[s + i] = S;

}

if(dir == D)

{

l[s] = '|';

}

print(root->left, s + len, U);

l[s] = '\0';

if(root->left == NULL && root->right == NULL)

{

buf[len - 1] = '\0';

printf("%s%s\n", l, buf);

}

else

{

printf("%s%s|\n", l, buf);

}

l[s] = S;

if(dir == U)

{

l[s] = '|';

}

print(root->right, s + len, D);

l[s] = S;

}

}

void printPre(Tree root, int s)

{

if(root != NULL)

{

int i;

char buf[10];

sprintf(buf, "%d-", root->key);

int len = strlen(buf);

for(i = 0; i < len; i++)

{

l[s + i] = S;

}

print(root->left, s + len, U);

printf("%s|\n", buf);

print(root->right, s + len, D);

}

}

int main(void)

{

int n;

Tree tree = NULL;

while(scanf("%d", &n) > 0)

{

Tree neo = malloc(sizeof(TNode));

neo->key = n;

neo->left = neo->right = NULL;

tree = insert(tree, neo);

}

printPre(tree, 0);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.StringTokenizer;

public class Main {

static int n;

static int[] a=new int[108];

static int[] left;

static int[] right;

static int[] level;

static int maxl=0;

static int M,N;

static char[][] map;

static boolean[] flag;

static boolean[] flag2;

static void add(int root,int index)

{

if(a[index]<a[root])

if(left[root]==0) { left[root]=index; level[index]=level[root]+1; if(level[index]>maxl) maxl=level[index]; }

else add(left[root],index);

else

if(right[root]==0) { right[root]=index; level[index]=level[root]+1; if(level[index]>maxl) maxl=level[index]; }

else add(right[root],index);

}

static void Fill(int root,int x,int y)

{

int lvl=level[root]+1;

String num=String.valueOf(a[root]);

int len=num.length();

flag[x]=true;

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

{

map[x][y-i]=num.charAt(i);

int h=2<<(maxl-lvl);

for(int j=1;j<h;j++) { map[x+j][y-i]='.'; map[x-j][y-i]='.'; }

}

if(left[root]==0 && right[root]==0 && level[root]<maxl) flag2[x]=true;

if(left[root]!=0)

Fill(left[root],x+(2<<(maxl-lvl))/2,y-8);

if(right[root]!=0)

Fill(right[root],x-(2<<(maxl-lvl))/2,y-8);

}

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

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

StringTokenizer tok=new StringTokenizer(bfr.readLine());

// long begin=System.currentTimeMillis();

int i,j;

for(i=1;tok.hasMoreTokens();i++) a[i]=Integer.parseInt(tok.nextToken());

n=i;

left =new int[n];

right =new int[n];

level=new int[n];

left[0]=1; right[0]=1; level[0]=0; level[1]=1;

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

add(1,i);

M=(2<<(maxl-1))-1;

N=8\*maxl-3;

map=new char[M][N];

flag=new boolean[M];

flag2=new boolean[M];

for(int l=1;l<maxl;l++)

{

int ty=l\*8-1;

int start=(2<<(l-1))-1,d=(2<<l);

for(i=0;i<M;i++) map[i][ty]=map[i][ty-1]=map[i][ty-2]='.';

for(i=start;i<M;i+=d)

{

map[i][ty]='-';

int hei=(2<<(l-1))/2;

for(int ii=0;ii<=hei;ii++)

{

map[i+ii][ty-1]='|';

map[i-ii][ty-1]='|';

}

map[i+hei][ty-2]='-';

map[i-hei][ty-2]='-';

}

}

Fill(1,M/2,N-1);

for(i=0;i<M;i++)

{

if(flag[i])

{

for(j=0;;j++) if(map[i][j]!=0 && map[i][j]!='.') break;

int jj;

for(jj=N-1;jj>j+1;jj--) if(map[i][jj]!=0) System.out.print(map[i][jj]);

if(!flag2[i]) System.out.print(map[i][jj]+""+map[i][jj-1]);

System.out.println();

}

}

// System.out.println(System.currentTimeMillis()-begin);

}

}

PREV-10 **历届试题 幸运数**

时间限制：1.0s   内存限制：256.0MB

关键字 堆

问题描述

幸运数是波兰数学家乌拉姆命名的。它采用与生成素数类似的“筛法”生成

。

首先从1开始写出自然数1,2,3,4,5,6,....

1 就是第一个幸运数。

我们从2这个数开始。把所有序号能被2整除的项删除，变为：

1 \_ 3 \_ 5 \_ 7 \_ 9 ....

把它们缩紧，重新记序，为：

1 3 5 7 9 .... 。这时，3为第2个幸运数，然后把所有能被3整除的序号位置的数删去。注意，是序号位置，不是那个数本身能否被3整除!! 删除的应该是5，11, 17, ...

此时7为第3个幸运数，然后再删去序号位置能被7整除的(19,39,...)

最后剩下的序列类似：

1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, ...

输入格式

输入两个正整数m n, 用空格分开 (m < n < 1000\*1000)

输出格式

程序输出 位于m和n之间的幸运数的个数（不包含m和n）。

样例输入1

1 20

样例输出1

5

样例输入2

30 69

样例输出2

8

**参考代码**

**C++**

#include <cstdio>

using namespace std;

int a[] = {1, 3, 7, 9, 13, 15, 21, 25, 31, 33, 37, 43, 49, 51, 63, 67, 69, 73, 75, 79, 87, 93, 99, 105, 111, 115, 127, 129, 133, 135, 141, 151, 159, 163, 169, 171, 189, 193, 195, 201, 205, 211, 219, 223, 231, 235, 237, 241, 259, 261, 267, 273, 283, 285, 289, 297, 303, 307, 319, 321, 327, 331, 339, 349, 357, 361, 367, 385, 391, 393, 399, 409, 415, 421, 427, 429, 433, 451, 463, 475, 477, 483, 487, 489, 495, 511, 517, 519, 529, 535, 537, 541, 553, 559, 577, 579, 583, 591, 601, 613, 615, 619, 621, 631, 639, 643, 645, 651, 655, 673, 679, 685, 693, 699, 717, 723, 727, 729, 735, 739, 741, 745, 769, 777, 781, 787, 801, 805, 819, 823, 831, 841, 855, 867, 873, 883, 885, 895, 897, 903, 925, 927, 931, 933, 937, 957, 961, 975, 979, 981, 991, 993, 997, 1009, 1011, 1021, 1023, 1029, 1039, 1041, 1053, 1057, 1087, 1093, 1095, 1101, 1105, 1107, 1117, 1123, 1147, 1155, 1167, 1179, 1183, 1189, 1197, 1201, 1203, 1209, 1219, 1231, 1233, 1245, 1249, 1251, 1261, 1263, 1275, 1281, 1285, 1291, 1303, 1309, 1323, 1329, 1339, 1357, 1365, 1369, 1387, 1389, 1395, 1401, 1417, 1419, 1435, 1441, 1455, 1459, 1471, 1473, 1485, 1491, 1495, 1497, 1501, 1503, 1519, 1533, 1543, 1545, 1563, 1567, 1575, 1579, 1585, 1587, 1597, 1599, 1611, 1639, 1641, 1645, 1659, 1663, 1675, 1693, 1701, 1705, 1711, 1723, 1731, 1737, 1749, 1765, 1767, 1771, 1773, 1777, 1797, 1801, 1809, 1819, 1827, 1831, 1833, 1839, 1857, 1869, 1879, 1893, 1899, 1915, 1921, 1923, 1933, 1941, 1945, 1959, 1963, 1965, 1977, 1983, 1987, 1995, 2001, 2019, 2023, 2031, 2053, 2059, 2065, 2067, 2079, 2083, 2085, 2095, 2113, 2115, 2121, 2125, 2127, 2133, 2163, 2173, 2187, 2209, 2211, 2215, 2217, 2221, 2239, 2251, 2253, 2257, 2271, 2277, 2281, 2283, 2301, 2311, 2317, 2323, 2335, 2343, 2355, 2365, 2379, 2395, 2403, 2407, 2409, 2415, 2419, 2427, 2439, 2445, 2461, 2467, 2473, 2479, 2491, 2493, 2505, 2511, 2523, 2527, 2545, 2557, 2563, 2571, 2575, 2587, 2589, 2593, 2599, 2607, 2625, 2635, 2647, 2649, 2653, 2661, 2667, 2671, 2689, 2697, 2715, 2725, 2755, 2763, 2773, 2781, 2785, 2787, 2797, 2815, 2817, 2821, 2823, 2827, 2835, 2841, 2845, 2851, 2877, 2887, 2899, 2901, 2905, 2913, 2923, 2943, 2953, 2961, 2971, 2973, 2977, 2983, 3003, 3007, 3027, 3031, 3037, 3039, 3049, 3055, 3073, 3075, 3091, 3097, 3099, 3109, 3111, 3121, 3123, 3133, 3153, 3163, 3171, 3175, 3183, 3187, 3199, 3213, 3223, 3229, 3235, 3243, 3259, 3261, 3289, 3297, 3301, 3307, 3313, 3325, 3339, 3351, 3355, 3363, 3381, 3403, 3405, 3409, 3411, 3427, 3433, 3439, 3451, 3453, 3465, 3477, 3481, 3487, 3489, 3495, 3499, 3507, 3559, 3565, 3571, 3579, 3595, 3597, 3603, 3607, 3613, 3621, 3625, 3633, 3655, 3661, 3663, 3669, 3675, 3685, 3687, 3697, 3709, 3717, 3721, 3727, 3747, 3753, 3763, 3771, 3781, 3789, 3793, 3795, 3811, 3843, 3847, 3849, 3865, 3873, 3879, 3889, 3891, 3897, 3909, 3915, 3931, 3943, 3951, 3955, 3969, 3975, 3981, 3991, 3999, 4003, 4015, 4023, 4033, 4035, 4041, 4045, 4063, 4069, 4081, 4095, 4105, 4107, 4129, 4131, 4161, 4165, 4173, 4179, 4189, 4195, 4201, 4203, 4207, 4227, 4237, 4251, 4255, 4257, 4263, 4269, 4285, 4287, 4315, 4321, 4329, 4335, 4363, 4377, 4383, 4389, 4399, 4413, 4431, 4441, 4443, 4455, 4461, 4465, 4483, 4485, 4495, 4509, 4519, 4521, 4539, 4551, 4561, 4567, 4569, 4573, 4587, 4609, 4611, 4621, 4623, 4629, 4645, 4647, 4653, 4663, 4671, 4675, 4695, 4699, 4713, 4717, 4725, 4741, 4761, 4767, 4773, 4797, 4801, 4809, 4813, 4819, 4833, 4837, 4839, 4843, 4851, 4863, 4867, 4881, 4887, 4893, 4929, 4951, 4963, 4965, 4969, 4977, 4987, 4989, 4993, 4999, 5001, 5007, 5019, 5029, 5041, 5043, 5049, 5053, 5089, 5103, 5127, 5137, 5139, 5149, 5151, 5157, 5169, 5179, 5181, 5191, 5211, 5217, 5229, 5233, 5235, 5253, 5259, 5277, 5283, 5293, 5295, 5299, 5325, 5335, 5341, 5343, 5371, 5377, 5379, 5385, 5409, 5419, 5427, 5433, 5449, 5455, 5463, 5473, 5487, 5491, 5503, 5515, 5527, 5547, 5551, 5559, 5569, 5577, 5587, 5589, 5593, 5599, 5613, 5617, 5637, 5641, 5649, 5655, 5661, 5671, 5673, 5679, 5691, 5701, 5707, 5713, 5719, 5737, 5755, 5763, 5767, 5769, 5803, 5809, 5817, 5827, 5833, 5839, 5851, 5869, 5883, 5889, 5893, 5901, 5905, 5911, 5913, 5923, 5959, 5965, 5967, 5971, 5973, 5977, 5991, 5997, 6009, 6019, 6031, 6049, 6055, 6061, 6079, 6093, 6111, 6115, 6123, 6141, 6147, 6159, 6163, 6175, 6177, 6195, 6211, 6229, 6237, 6243, 6249, 6253, 6271, 6273, 6279, 6301, 6309, 6331, 6345, 6351, 6355, 6363, 6367, 6369, 6373, 6379, 6385, 6399, 6411, 6415, 6427, 6433, 6435, 6447, 6463, 6471, 6475, 6477, 6501, 6505, 6523, 6531, 6535, 6541, 6553, 6559, 6567, 6573, 6601, 6621, 6625, 6631, 6661, 6663, 6667, 6669, 6679, 6687, 6693, 6715, 6723, 6733, 6741, 6745, 6747, 6753, 6757, 6763, 6765, 6783, 6787, 6789, 6841, 6849, 6867, 6871, 6883, 6891, 6909, 6915, 6921, 6931, 6933, 6937, 6951, 6981, 6985, 6999, 7003, 7009, 7035, 7041, 7045, 7047, 7069, 7077, 7081, 7087, 7101, 7111, 7129, 7135, 7153, 7167, 7171, 7173, 7183, 7191, 7195, 7197, 7207, 7215, 7231, 7233, 7237, 7245, 7249, 7279, 7293, 7299, 7311, 7321, 7333, 7339, 7341, 7345, 7357, 7359, 7371, 7377, 7395, 7401, 7405, 7419, 7435, 7437, 7443, 7447, 7459, 7471, 7489, 7501, 7503, 7507, 7531, 7533, 7549, 7551, 7563, 7567, 7575, 7585, 7591, 7593, 7603, 7609, 7629, 7633, 7639, 7645, 7677, 7687, 7689, 7701, 7711, 7717, 7737, 7755, 7779, 7791, 7795, 7801, 7803, 7813, 7827, 7833, 7837, 7855, 7881, 7885, 7897, 7899, 7909, 7917, 7921, 7929, 7939, 7947, 7951, 7963, 7969, 7971, 8001, 8005, 8007, 8013, 8037, 8047, 8071, 8073, 8085, 8089, 8107, 8109, 8127, 8131, 8139, 8151, 8161, 8169, 8173, 8175, 8191, 8193, 8221, 8223, 8227, 8233, 8253, 8257, 8263, 8269, 8283, 8289, 8299, 8325, 8331, 8347, 8359, 8365, 8367, 8379, 8409, 8413, 8421, 8445, 8449, 8467, 8473, 8475, 8479, 8487, 8493, 8515, 8535, 8539, 8551, 8553, 8575, 8577, 8583, 8589, 8601, 8605, 8611, 8617, 8635, 8637, 8641, 8647, 8655, 8661, 8673, 8683, 8719, 8757, 8769, 8773, 8787, 8793, 8805, 8809, 8827, 8829, 8833, 8835, 8841, 8869, 8871, 8895, 8913, 8931, 8937, 8947, 8955, 8961, 8977, 8979, 8989, 8995, 8997, 9009, 9031, 9033, 9061, 9063, 9073, 9075, 9081, 9085, 9115, 9117, 9121, 9123, 9135, 9139, 9145, 9151, 9177, 9181, 9211, 9213, 9223, 9231, 9235, 9249, 9253, 9267, 9273, 9277, 9285, 9291, 9303, 9315, 9331, 9339, 9349, 9373, 9387, 9399, 9403, 9409, 9421, 9423, 9429, 9441, 9451, 9457, 9471, 9475, 9481, 9483, 9487, 9501, 9535, 9537, 9543, 9547, 9549, 9555, 9561, 9567, 9613, 9621, 9625, 9631, 9643, 9649, 9661, 9663, 9675, 9687, 9691, 9703, 9727, 9733, 9751, 9753, 9775, 9787, 9789, 9795, 9801, 9807, 9811, 9813, 9837, 9841, 9883, 9895, 9915, 9927, 9937, 9957, 9961, 9979, 9985, 9987, 9997, 9999, 10003, 10009, 10017, 10041, 10051, 10059, 10069, 10071, 10083, 10087, 10089, 10095, 10111, 10117, 10125, 10131, 10143, 10147, 10149, 10153, 10173, 10191, 10195, 10197, 10221, 10239, 10255, 10261, 10275, 10279, 10291, 10311, 10315, 10317, 10321, 10341, 10363, 10365, 10387, 10399, 10411, 10417, 10431, 10441, 10443, 10459, 10461, 10471, 10501, 10507, 10509, 10525, 10531, 10533, 10555, 10563, 10569, 10575, 10585, 10587, 10597, 10599, 10605, 10609, 10635, 10651, 10653, 10659, 10671, 10683, 10723, 10725, 10731, 10747, 10753, 10755, 10759, 10773, 10785, 10789, 10797, 10809, 10843, 10849, 10873, 10881, 10891, 10903, 10909, 10911, 10921, 10923, 10945, 10953, 10965, 10975, 10977, 11005, 11011, 11017, 11047, 11055, 11059, 11073, 11077, 11089, 11091, 11097, 11113, 11137, 11157, 11163, 11173, 11181, 11193, 11197, 11205, 11217, 11223, 11227, 11239, 11245, 11247, 11259, 11263, 11269, 11283, 11289, 11301, 11323, 11341, 11343, 11371, 11373, 11377, 11389, 11391, 11395, 11407, 11419, 11425, 11427, 11437, 11439, 11475, 11479, 11487, 11491, 11521, 11529, 11535, 11539, 11541, 11577, 11581, 11583, 11599, 11617, 11623, 11629, 11641, 11643, 11659, 11667, 11677, 11679, 11707, 11721, 11725, 11731, 11769, 11781, 11797, 11803, 11811, 11815, 11823, 11827, 11833, 11835, 11845, 11847, 11853, 11865, 11877, 11887, 11911, 11923, 11929, 11943, 11953, 11959, 11961, 11991, 11995, 12007, 12015, 12025, 12049, 12057, 12063, 12069, 12079, 12097, 12103, 12117, 12121, 12133, 12139, 12141, 12151, 12159, 12163, 12165, 12193, 12201, 12207, 12223, 12237, 12243, 12265, 12267, 12285, 12295, 12301, 12307, 12315, 12319, 12321, 12333, 12337, 12357, 12363, 12373, 12379, 12391, 12399, 12411, 12417, 12427, 12435, 12457, 12481, 12487, 12495, 12499, 12519, 12543, 12547, 12553, 12567, 12579, 12589, 12607, 12609, 12621, 12631, 12645, 12649, 12651, 12673, 12679, 12699, 12709, 12711, 12729, 12735, 12741, 12759, 12763, 12777, 12781, 12799, 12801, 12813, 12819, 12861, 12867, 12877, 12885, 12903, 12907, 12925, 12937, 12951, 12957, 12961, 12969, 12991, 12999, 13009, 13011, 13021, 13027, 13029, 13053, 13057, 13063, 13077, 13083, 13113, 13117, 13119, 13125, 13129, 13135, 13141, 13167, 13177, 13179, 13203, 13209, 13215, 13219, 13243, 13245, 13267, 13273, 13279, 13281, 13285, 13293, 13309, 13317, 13323, 13327, 13335, 13345, 13347, 13359, 13363, 13371, 13387, 13419, 13441, 13443, 13465, 13473, 13485, 13489, 13513, 13515, 13519, 13527, 13531, 13533, 13537, 13551, 13567, 13575, 13581, 13587, 13599, 13609, 13633, 13657, 13677, 13681, 13693, 13695, 13707, 13723, 13725, 13737, 13741, 13743, 13759, 13801, 13803, 13821, 13831, 13833, 13849, 13851, 13861, 13873, 13893, 13897, 13929, 13947, 13953, 13957, 13995, 14007, 14017, 14031, 14035, 14037, 14041, 14049, 14053, 14059, 14073, 14079, 14091, 14095, 14113, 14115, 14119, 14137, 14163, 14179, 14181, 14185, 14199, 14205, 14209, 14211, 14217, 14229, 14245, 14247, 14253, 14281, 14287, 14293, 14307, 14313, 14317, 14331, 14347, 14349, 14353, 14367, 14395, 14407, 14419, 14427, 14437, 14439, 14443, 14449, 14457, 14461, 14473, 14491, 14493, 14505, 14521, 14523, 14539, 14541, 14565, 14587, 14595, 14599, 14601, 14607, 14625, 14661, 14667, 14689, 14691, 14713, 14715, 14721, 14731, 14749, 14751, 14755, 14775, 14779, 14785, 14793, 14797, 14817, 14841, 14875, 14881, 14883, 14899, 14905, 14919, 14923, 14931, 14935, 14943, 14947, 14973, 14977, 14997, 15003, 15007, 15009, 15025, 15031, 15049, 15057, 15063, 15069, 15073, 15081, 15103, 15109, 15121, 15129, 15133, 15141, 15165, 15169, 15175, 15183, 15187, 15189, 15193, 15205, 15213, 15217, 15229, 15231, 15235, 15237, 15247, 15255, 15259, 15261, 15271, 15291, 15301, 15313, 15333, 15355, 15379, 15381, 15385, 15397, 15417, 15445, 15457, 15469, 15471, 15481, 15487, 15489, 15511, 15519, 15523, 15531, 15543, 15547, 15567, 15573, 15577, 15585, 15597, 15609, 15645, 15655, 15661, 15673, 15675, 15693, 15697, 15717, 15735, 15763, 15771, 15783, 15793, 15805, 15817, 15819, 15835, 15843, 15847, 15855, 15879, 15883, 15891, 15901, 15907, 15919, 15927, 15939, 15955, 15969, 15975, 15981, 15993, 16003, 16009, 16023, 16045, 16047, 16053, 16057, 16077, 16081, 16095, 16101, 16135, 16153, 16191, 16195, 16197, 16203, 16213, 16227, 16237, 16245, 16255, 16267, 16275, 16279, 16287, 16297, 16309, 16329, 16333, 16341, 16353, 16359, 16365, 16381, 16387, 16389, 16393, 16395, 16413, 16417, 16429, 16431, 16447, 16449, 16467, 16473, 16479, 16491, 16507, 16521, 16539, 16557, 16573, 16591, 16611, 16615, 16617, 16621, 16635, 16639, 16657, 16665, 16677, 16701, 16707, 16725, 16747, 16765, 16771, 16773, 16791, 16795, 16825, 16827, 16845, 16863, 16867, 16893, 16915, 16917, 16921, 16933, 16939, 16947, 16953, 16957, 16959, 16969, 16971, 16981, 16995, 17001, 17011, 17017, 17019, 17035, 17047, 17053, 17061, 17083, 17085, 17103, 17115, 17119, 17125, 17127, 17151, 17157, 17173, 17181, 17209, 17211, 17215, 17221, 17223, 17229, 17251, 17253, 17283, 17287, 17305, 17313, 17329, 17335, 17341, 17355, 17359, 17367, 17389, 17391, 17401, 17409, 17431, 17433, 17455, 17463, 17467, 17473, 17485, 17497, 17517, 17527, 17535, 17545, 17547, 17563, 17565, 17581, 17589, 17619, 17625, 17671, 17677, 17689, 17695, 17703, 17713, 17725, 17733, 17737, 17749, 17767, 17769, 17775, 17781, 17787, 17797, 17811, 17829, 17851, 17875, 17883, 17893, 17913, 17925, 17935, 17943, 17947, 17965, 17977, 17991, 17997, 18001, 18007, 18019, 18025, 18043, 18049, 18055, 18067, 18085, 18087, 18123, 18127, 18145, 18147, 18151, 18153, 18169, 18175, 18181, 18187, 18189, 18195, 18207, 18211, 18237, 18253, 18261, 18279, 18283, 18285, 18295, 18303, 18315, 18333, 18375, 18381, 18397, 18399, 18403, 18409, 18441, 18445, 18451, 18459, 18465, 18469, 18475, 18493, 18495, 18501, 18513, 18523, 18535, 18543, 18589, 18591, 18597, 18615, 18619, 18621, 18633, 18639, 18655, 18661, 18681, 18693, 18715, 18717, 18723, 18745, 18747, 18757, 18759, 18765, 18775, 18777, 18783, 18787, 18817, 18819, 18825, 18837, 18849, 18853, 18871, 18873, 18879, 18915, 18937, 18943, 18951, 18967, 18969, 18975, 18979, 18987, 19009, 19015, 19017, 19033, 19051, 19081, 19113, 19119, 19125, 19131, 19153, 19155, 19159, 19161, 19173, 19195, 19197, 19207, 19225, 19237, 19261, 19263, 19269, 19285, 19309, 19323, 19327, 19329, 19333, 19341, 19347, 19351, 19357, 19363, 19365, 19371, 19389, 19395, 19405, 19411, 19417, 19435, 19455, 19467, 19477, 19479, 19503, 19521, 19533, 19543, 19551, 19579, 19581, 19603, 19609, 19615, 19627, 19629, 19657, 19659, 19687, 19693, 19705, 19707, 19711, 19719, 19723, 19725, 19731, 19735, 19755, 19767, 19771, 19773, 19783, 19791, 19807, 19825, 19851, 19861, 19869, 19875, 19893, 19917, 19921, 19929, 19939, 19963, 19981, 19993, 20005, 20013, 20017, 20043, 20067, 20071, 20077, 20085, 20089, 20101, 20103, 20121, 20133, 20149, 20151, 20169, 20181, 20185, 20197, 20203, 20215, 20227, 20253, 20257, 20271, 20295, 20307, 20319, 20341, 20349, 20353, 20355, 20359, 20371, 20379, 20385, 20397, 20403, 20425, 20445, 20449, 20455, 20461, 20475, 20485, 20487, 20509, 20511, 20539, 20551, 20563, 20581, 20583, 20587, 20605, 20611, 20613, 20649, 20665, 20679, 20697, 20713, 20731, 20733, 20739, 20757, 20763, 20769, 20773, 20793, 20797, 20805, 20811, 20821, 20823, 20833, 20835, 20839, 20845, 20859, 20875, 20887, 20889, 20899, 20905, 20923, 20937, 20941, 20953, 20959, 20965, 20991, 21013, 21015, 21021, 21027, 21043, 21049, 21051, 21055, 21073, 21075, 21111, 21115, 21129, 21139, 21147, 21153, 21169, 21175, 21177, 21189, 21193, 21199, 21205, 21211, 21223, 21231, 21253, 21255, 21307, 21309, 21315, 21327, 21331, 21337, 21345, 21373, 21381, 21405, 21409, 21411, 21421, 21433, 21445, 21451, 21453, 21469, 21471, 21475, 21499, 21505, 21529, 21531, 21537, 21549, 21567, 21571, 21591, 21597, 21609, 21613, 21615, 21621, 21655, 21663, 21687, 21693, 21703, 21709, 21717, 21721, 21735, 21739, 21751, 21757, 21759, 21771, 21787, 21819, 21841, 21847, 21867, 21885, 21891, 21897, 21913, 21915, 21925, 21927, 21931, 21933, 21957, 21967, 21987, 21993, 22009, 22017, 22035, 22039, 22075, 22087, 22093, 22095, 22099, 22113, 22119, 22125, 22147, 22155, 22165, 22167, 22177, 22185, 22197, 22225, 22227, 22231, 22269, 22281, 22305, 22327, 22333, 22339, 22369, 22371, 22377, 22381, 22399, 22401, 22411, 22417, 22431, 22437, 22461, 22471, 22473, 22479, 22503, 22507, 22521, 22525, 22527, 22533, 22555, 22569, 22585, 22587, 22591, 22597, 22603, 22617, 22627, 22629, 22651, 22683, 22687, 22689, 22701, 22711, 22743, 22779, 22789, 22791, 22795, 22807, 22827, 22839, 22843, 22851, 22855, 22869, 22879, 22881, 22885, 22893, 22911, 22921, 22941, 22947, 22953, 22957, 22965, 22969, 22975, 22981, 22987, 22999, 23007, 23031, 23037, 23041, 23061, 23067, 23083, 23089, 23095, 23101, 23109, 23137, 23145, 23163, 23169, 23173, 23191, 23193, 23199, 23205, 23269, 23271, 23293, 23295, 23311, 23323, 23341, 23343, 23353, 23355, 23361, 23365, 23373, 23377, 23397, 23419, 23421, 23427, 23445, 23449, 23451, 23457, 23469, 23479, 23499, 23515, 23523, 23529, 23533, 23551, 23563, 23565, 23569, 23571, 23577, 23607, 23613, 23649, 23671, 23677, 23689, 23701, 23709, 23721, 23767, 23775, 23781, 23799, 23805, 23815, 23829, 23835, 23851, 23857, 23881, 23889, 23893, 23913, 23919, 23925, 23929, 23943, 23953, 23955, 23961, 23965, 23971, 23973, 23991, 24013, 24015, 24025, 24033, 24037, 24069, 24073, 24117, 24121, 24129, 24133, 24145, 24151, 24181, 24183, 24213, 24217, 24237, 24247, 24255, 24261, 24277, 24285, 24289, 24301, 24309, 24321, 24327, 24331, 24333, 24343, 24363, 24381, 24387, 24397, 24403, 24405, 24423, 24429, 24447, 24451, 24481, 24499, 24523, 24531, 24571, 24573, 24585, 24591, 24603, 24619, 24633, 24637, 24639, 24643, 24645, 24649, 24663, 24669, 24709, 24711, 24727, 24729, 24733, 24739, 24741, 24759, 24763, 24771, 24783, 24789, 24805, 24811, 24829, 24831, 24843, 24855, 24865, 24873, 24877, 24895, 24907, 24909, 24933, 24951, 24957, 24963, 24985, 24991, 25015, 25017, 25021, 25033, 25057, 25063, 25095, 25099, 25107, 25117, 25119, 25129, 25147, 25153, 25159, 25167, 25171, 25185, 25189, 25201, 25213, 25215, 25225, 25237, 25255, 25263, 25285, 25287, 25293, 25309, 25317, 25335, 25341, 25347, 25377, 25389, 25413, 25425, 25431, 25441, 25455, 25461, 25477, 25483, 25489, 25497, 25503, 25525, 25549, 25561, 25567, 25569, 25581, 25591, 25593, 25611, 25621, 25623, 25645, 25647, 25657, 25671, 25677, 25687, 25737, 25753, 25759, 25771, 25773, 25803, 25815, 25819, 25821, 25833, 25837, 25845, 25855, 25861, 25875, 25893, 25909, 25917, 25923, 25929, 25939, 25959, 25963, 25971, 25977, 25987, 26005, 26007, 26011, 26025, 26035, 26041, 26071, 26085, 26103, 26107, 26115, 26119, 26149, 26155, 26167, 26169, 26179, 26209, 26215, 26217, 26221, 26229, 26251, 26263, 26281, 26293, 26295, 26301, 26313, 26319, 26343, 26347, 26359, 26365, 26385, 26397, 26409, 26413, 26431, 26445, 26467, 26505, 26523, 26529, 26533, 26547, 26553, 26565, 26595, 26599, 26617, 26629, 26641, 26649, 26655, 26659, 26665, 26683, 26695, 26697, 26703, 26719, 26721, 26727, 26733, 26749, 26755, 26757, 26781, 26797, 26805, 26841, 26845, 26847, 26881, 26883, 26893, 26905, 26907, 26913, 26923, 26925, 26935, 26943, 26997, 27001, 27007, 27009, 27013, 27027, 27031, 27033, 27037, 27039, 27061, 27063, 27075, 27079, 27091, 27093, 27097, 27105, 27141, 27157, 27189, 27195, 27199, 27201, 27205, 27207, 27217, 27237, 27265, 27271, 27283, 27301, 27309, 27315, 27321, 27325, 27331, 27333, 27349, 27357, 27363, 27387, 27393, 27409, 27415, 27417, 27427, 27435, 27453, 27457, 27469, 27475, 27477, 27483, 27489, 27493, 27501, 27511, 27531, 27541, 27561, 27585, 27595, 27597, 27603, 27639, 27663, 27667, 27691, 27693, 27705, 27711, 27741, 27751, 27757, 27765, 27783, 27793, 27805, 27807, 27813, 27829, 27831, 27847, 27855, 27859, 27871, 27879, 27913, 27915, 27951, 27961, 27963, 27975, 28021, 28023, 28027, 28039, 28069, 28071, 28083, 28087, 28089, 28101, 28105, 28119, 28123, 28129, 28131, 28141, 28161, 28167, 28191, 28195, 28203, 28233, 28239, 28245, 28293, 28311, 28321, 28329, 28339, 28351, 28359, 28365, 28375, 28393, 28405, 28417, 28429, 28435, 28437, 28459, 28467, 28483, 28509, 28513, 28525, 28563, 28569, 28581, 28593, 28605, 28633, 28639, 28677, 28689, 28701, 28707, 28735, 28759, 28765, 28771, 28791, 28797, 28801, 28803, 28815, 28825, 28827, 28837, 28843, 28855, 28857, 28867, 28869, 28887, 28891, 28905, 28921, 28923, 28933, 28951, 28963, 28969, 28987, 28989, 28993, 28995, 29005, 29025, 29031, 29035, 29049, 29055, 29065, 29089, 29091, 29095, 29113, 29121, 29127, 29137, 29143, 29149, 29157, 29175, 29185, 29193, 29203, 29205, 29239, 29253, 29281, 29283, 29287, 29295, 29311, 29317, 29319, 29329, 29341, 29349, 29371, 29391, 29413, 29437, 29451, 29457, 29467, 29473, 29485, 29487, 29493, 29497, 29515, 29517, 29521, 29533, 29535, 29563, 29577, 29589, 29595, 29599, 29611, 29613, 29623, 29625, 29653, 29659, 29661, 29665, 29677, 29689, 29697, 29709, 29715, 29721, 29751, 29757, 29779, 29781, 29787, 29791, 29805, 29815, 29829, 29845, 29847, 29851, 29865, 29877, 29893, 29905, 29913, 29925, 29949, 29959, 29971, 29973, 29989, 29995, 30013, 30021, 30061, 30067, 30073, 30075, 30087, 30103, 30115, 30121, 30159, 30163, 30181, 30207, 30225, 30229, 30231, 30241, 30243, 30253, 30261, 30265, 30273, 30289, 30291, 30303, 30313, 30327, 30337, 30367, 30369, 30373, 30387, 30397, 30409, 30411, 30429, 30439, 30465, 30471, 30475, 30477, 30495, 30499, 30505, 30523, 30529, 30537, 30541, 30543, 30547, 30559, 30565, 30567, 30589, 30601, 30643, 30651, 30681, 30687, 30727, 30747, 30753, 30765, 30775, 30777, 30789, 30793, 30807, 30817, 30837, 30849, 30861, 30871, 30873, 30877, 30885, 30901, 30915, 30919, 30943, 30945, 30949, 30969, 30997, 30999, 31003, 31021, 31029, 31033, 31039, 31051, 31069, 31071, 31081, 31093, 31095, 31107, 31135, 31137, 31143, 31147, 31189, 31191, 31207, 31221, 31231, 31237, 31239, 31249, 31251, 31255, 31261, 31263, 31293, 31303, 31311, 31315, 31335, 31341, 31363, 31381, 31383, 31387, 31389, 31395, 31405, 31411, 31425, 31429, 31437, 31449, 31453, 31489, 31491, 31515, 31531, 31545, 31573, 31611, 31617, 31633, 31639, 31641, 31681, 31699, 31701, 31713, 31723, 31737, 31755, 31765, 31773, 31789, 31801, 31807, 31815, 31839, 31849, 31863, 31869, 31879, 31887, 31893, 31903, 31927, 31945, 31953, 31965, 31971, 31975, 31977, 31989, 32005, 32011, 32013, 32019, 32041, 32053, 32055, 32077, 32079, 32097, 32101, 32115, 32119, 32121, 32133, 32145, 32173, 32179, 32197, 32223, 32229, 32245, 32247, 32265, 32269, 32271, 32277, 32287, 32311, 32325, 32329, 32373, 32385, 32391, 32397, 32403, 32407, 32419, 32433, 32455, 32457, 32467, 32481, 32509, 32533, 32539, 32545, 32563, 32571, 32577, 32593, 32595, 32605, 32607, 32613, 32635, 32637, 32641, 32643, 32665, 32683, 32713, 32727, 32731, 32745, 32761, 32769, 32775, 32793, 32803, 32827, 32829, 32833, 32845, 32853, 32865, 32869, 32875, 32889, 32917, 32919, 32923, 32937, 32961, 32973, 32985, 32997, 33019, 33021, 33027, 33033, 33045, 33067, 33079, 33091, 33105, 33109, 33117, 33121, 33141, 33151, 33153, 33169, 33181, 33183, 33189, 33205, 33211, 33217, 33237, 33247, 33249, 33273, 33289, 33297, 33309, 33313, 33327, 33333, 33337, 33349, 33363, 33375, 33379, 33381, 33411, 33423, 33433, 33435, 33439, 33445, 33463, 33465, 33475, 33495, 33525, 33537, 33541, 33549, 33561, 33567, 33595, 33601, 33603, 33615, 33633, 33645, 33657, 33673, 33687, 33705, 33717, 33721, 33757, 33769, 33781, 33805, 33813, 33835, 33837, 33843, 33855, 33861, 33885, 33895, 33897, 33907, 33909, 33925, 33943, 33949, 33961, 33973, 33979, 33987, 33991, 33993, 34003, 34005, 34021, 34027, 34053, 34065, 34093, 34099, 34105, 34107, 34113, 34125, 34129, 34131, 34161, 34171, 34183, 34191, 34197, 34201, 34209, 34213, 34233, 34257, 34261, 34281, 34293, 34305, 34321, 34323, 34357, 34359, 34365, 34377, 34387, 34399, 34401, 34423, 34435, 34449, 34453, 34461, 34473, 34485, 34491, 34497, 34503, 34507, 34513, 34557, 34569, 34573, 34599, 34603, 34617, 34621, 34623, 34639, 34641, 34653, 34687, 34713, 34717, 34719, 34723, 34725, 34737, 34755, 34761, 34767, 34777, 34785, 34807, 34809, 34813, 34821, 34873, 34875, 34885, 34887, 34891, 34909, 34915, 34917, 34951, 34957, 34965, 34971, 34995, 35011, 35035, 35041, 35061, 35071, 35077, 35091, 35103, 35113, 35127, 35137, 35145, 35155, 35167, 35169, 35175, 35197, 35203, 35205, 35221, 35223, 35233, 35287, 35301, 35317, 35329, 35331, 35343, 35349, 35391, 35395, 35409, 35413, 35419, 35421, 35427, 35443, 35449, 35451, 35457, 35461, 35481, 35485, 35511, 35539, 35547, 35557, 35563, 35577, 35587, 35601, 35607, 35611, 35625, 35643, 35673, 35683, 35701, 35703, 35707, 35709, 35713, 35721, 35743, 35799, 35817, 35833, 35847, 35859, 35871, 35881, 35913, 35917, 35919, 35941, 35943, 35959, 35965, 35973, 36003, 36025, 36037, 36039, 36043, 36051, 36067, 36085, 36087, 36099, 36115, 36129, 36135, 36147, 36151, 36165, 36169, 36175, 36187, 36205, 36213, 36231, 36249, 36255, 36261, 36273, 36277, 36297, 36309, 36319, 36325, 36331, 36351, 36361, 36373, 36375, 36381, 36405, 36415, 36417, 36423, 36427, 36429, 36451, 36469, 36481, 36487, 36499, 36501, 36513, 36529, 36541, 36547, 36549, 36561, 36573, 36583, 36603, 36609, 36613, 36615, 36619, 36633, 36669, 36673, 36675, 36679, 36681, 36687, 36691, 36715, 36721, 36733, 36741, 36759, 36763, 36805, 36823, 36825, 36829, 36837, 36855, 36877, 36891, 36897, 36903, 36907, 36943, 36955, 36963, 36987, 36993, 36997, 37011, 37015, 37023, 37047, 37051, 37069, 37075, 37077, 37081, 37087, 37089, 37095, 37117, 37129, 37161, 37185, 37195, 37203, 37215, 37219, 37239, 37255, 37263, 37267, 37279, 37285, 37287, 37317, 37339, 37345, 37365, 37371, 37383, 37389, 37393, 37395, 37435, 37437, 37447, 37455, 37467, 37491, 37495, 37507, 37509, 37521, 37527, 37531, 37533, 37549, 37569, 37585, 37597, 37599, 37621, 37635, 37663, 37683, 37689, 37695, 37705, 37723, 37729, 37737, 37747, 37749, 37753, 37789, 37801, 37809, 37813, 37815, 37833, 37845, 37851, 37855, 37893, 37905, 37909, 37939, 37941, 37957, 37959, 37963, 37969, 37989, 38001, 38031, 38035, 38037, 38041, 38059, 38067, 38097, 38119, 38121, 38131, 38145, 38149, 38167, 38185, 38209, 38221, 38245, 38247, 38265, 38271, 38305, 38307, 38313, 38319, 38325, 38341, 38353, 38355, 38389, 38391, 38397, 38401, 38409, 38421, 38433, 38439, 38455, 38461, 38467, 38473, 38481, 38503, 38527, 38535, 38559, 38563, 38581, 38599, 38611, 38623, 38625, 38629, 38653, 38661, 38685, 38695, 38707, 38713, 38731, 38737, 38749, 38751, 38757, 38769, 38775, 38779, 38787, 38791, 38797, 38799, 38811, 38815, 38821, 38839, 38851, 38853, 38871, 38875, 38913, 38917, 38923, 38925, 38943, 38949, 38967, 38985, 38989, 39007, 39043, 39045, 39061, 39067, 39073, 39093, 39115, 39123, 39135, 39139, 39159, 39171, 39177, 39193, 39235, 39249, 39265, 39271, 39283, 39285, 39295, 39301, 39315, 39321, 39345, 39355, 39361, 39367, 39387, 39411, 39417, 39441, 39451, 39463, 39475, 39481, 39507, 39513, 39553, 39567, 39571, 39573, 39579, 39585, 39589, 39607, 39609, 39615, 39627, 39633, 39637, 39643, 39649, 39673, 39693, 39705, 39711, 39721, 39733, 39741, 39753, 39769, 39783, 39795, 39801, 39825, 39831, 39837, 39841, 39849, 39861, 39865, 39883, 39889, 39891, 39903, 39909, 39931, 39945, 39955, 39957, 39967, 39979, 39987, 40005, 40029, 40041, 40059, 40069, 40083, 40101, 40117, 40141, 40147, 40153, 40161, 40165, 40167, 40195, 40197, 40201, 40207, 40215, 40225, 40231, 40249, 40257, 40273, 40287, 40309, 40321, 40323, 40333, 40335, 40351, 40353, 40369, 40389, 40393, 40399, 40405, 40407, 40447, 40449, 40453, 40455, 40459, 40471, 40477, 40479, 40515, 40521, 40555, 40561, 40593, 40597, 40603, 40609, 40665, 40671, 40701, 40707, 40713, 40729, 40743, 40747, 40753, 40761, 40765, 40767, 40797, 40803, 40815, 40825, 40831, 40837, 40849, 40855, 40861, 40879, 40911, 40921, 40923, 40929, 40933, 40963, 40981, 40983, 40993, 40999, 41013, 41023, 41025, 41029, 41043, 41055, 41089, 41101, 41107, 41119, 41121, 41151, 41161, 41181, 41185, 41191, 41235, 41245, 41247, 41277, 41281, 41287, 41295, 41307, 41313, 41319, 41331, 41341, 41349, 41365, 41373, 41383, 41395, 41401, 41421, 41425, 41439, 41463, 41469, 41475, 41485, 41503, 41527, 41529, 41539, 41541, 41551, 41563, 41595, 41613, 41631, 41635, 41653, 41655, 41667, 41677, 41691, 41697, 41707, 41715, 41721, 41727, 41731, 41743, 41779, 41781, 41791, 41793, 41799, 41803, 41805, 41823, 41845, 41847, 41853, 41869, 41905, 41925, 41937, 41943, 41947, 41949, 41961, 41989, 42003, 42009, 42025, 42033, 42051, 42055, 42063, 42073, 42091, 42109, 42117, 42129, 42153, 42163, 42169, 42189, 42193, 42201, 42211, 42231, 42235, 42243, 42279, 42303, 42309, 42361, 42367, 42373, 42381, 42391, 42403, 42405, 42421, 42435, 42447, 42451, 42469, 42471, 42495, 42499, 42511, 42517, 42525, 42547, 42555, 42559, 42561, 42567, 42577, 42591, 42609, 42619, 42625, 42643, 42651, 42657, 42661, 42663, 42675, 42693, 42705, 42715, 42727, 42739, 42751, 42759, 42781, 42789, 42813, 42829, 42847, 42849, 42853, 42865, 42873, 42883, 42885, 42895, 42903, 42907, 42913, 42933, 42949, 42955, 42967, 42975, 42979, 43015, 43021, 43029, 43035, 43039, 43071, 43077, 43129, 43135, 43159, 43167, 43185, 43191, 43201, 43207, 43209, 43225, 43227, 43231, 43233, 43239, 43251, 43255, 43269, 43285, 43303, 43315, 43329, 43333, 43345, 43347, 43381, 43407, 43413, 43419, 43423, 43429, 43441, 43443, 43459, 43461, 43479, 43483, 43491, 43495, 43507, 43513, 43515, 43519, 43533, 43543, 43545, 43549, 43557, 43579, 43597, 43621, 43633, 43639, 43669, 43681, 43695, 43711, 43729, 43735, 43743, 43747, 43771, 43789, 43797, 43815, 43819, 43831, 43833, 43839, 43857, 43863, 43873, 43879, 43891, 43899, 43923, 43933, 43953, 43957, 43965, 43975, 43977, 43981, 43983, 43995, 44007, 44023, 44025, 44037, 44049, 44059, 44109, 44145, 44149, 44155, 44163, 44175, 44187, 44197, 44229, 44239, 44241, 44259, 44293, 44295, 44311, 44313, 44319, 44341, 44353, 44355, 44361, 44383, 44385, 44389, 44395, 44401, 44415, 44419, 44421, 44451, 44485, 44487, 44499, 44503, 44511, 44515, 44541, 44545, 44547, 44551, 44563, 44571, 44595, 44625, 44635, 44637, 44641, 44649, 44655, 44659, 44673, 44691, 44697, 44703, 44721, 44731, 44751, 44761, 44775, 44805, 44863, 44869, 44889, 44893, 44911, 44923, 44929, 44931, 44953, 44955, 44961, 44983, 44995, 45013, 45051, 45057, 45069, 45081, 45109, 45111, 45121, 45133, 45139, 45145, 45159, 45171, 45175, 45181, 45183, 45201, 45235, 45241, 45267, 45271, 45277, 45279, 45303, 45307, 45327, 45331, 45343, 45349, 45351, 45363, 45373, 45375, 45381, 45393, 45403, 45429, 45435, 45459, 45469, 45489, 45507, 45519, 45529, 45535, 45541, 45555, 45591, 45603, 45613, 45619, 45621, 45625, 45681, 45685, 45709, 45717, 45721, 45723, 45739, 45745, 45763, 45769, 45789, 45801, 45811, 45813, 45831, 45843, 45877, 45897, 45907, 45919, 45927, 45939, 45951, 45957, 45973, 45975, 45981, 45991, 46015, 46021, 46023, 46027, 46035, 46065, 46069, 46087, 46095, 46101, 46129, 46131, 46137, 46141, 46167, 46195, 46201, 46203, 46213, 46221, 46227, 46231, 46243, 46245, 46255, 46257, 46263, 46279, 46309, 46311, 46327, 46329, 46339, 46341, 46369, 46377, 46389, 46399, 46401, 46405, 46461, 46465, 46497, 46503, 46519, 46527, 46531, 46543, 46549, 46557, 46587, 46599, 46629, 46641, 46651, 46653, 46671, 46683, 46687, 46695, 46705, 46707, 46713, 46717, 46735, 46737, 46749, 46755, 46759, 46791, 46795, 46797, 46801, 46813, 46815, 46839, 46845, 46855, 46861, 46875, 46879, 46887, 46915, 46923, 46939, 46947, 46951, 46965, 46987, 46999, 47001, 47005, 47029, 47047, 47065, 47071, 47073, 47103, 47107, 47109, 47115, 47125, 47127, 47155, 47157, 47167, 47179, 47193, 47221, 47229, 47233, 47235, 47265, 47271, 47281, 47287, 47313, 47319, 47329, 47335, 47349, 47361, 47379, 47383, 47385, 47419, 47421, 47425, 47427, 47439, 47443, 47449, 47461, 47463, 47487, 47505, 47523, 47565, 47571, 47575, 47581, 47607, 47611, 47613, 47629, 47631, 47665, 47677, 47683, 47697, 47703, 47715, 47727, 47739, 47743, 47745, 47757, 47775, 47799, 47823, 47833, 47841, 47847, 47865, 47889, 47895, 47901, 47911, 47913, 47923, 47947, 47949, 47959, 47965, 47989, 48007, 48015, 48019, 48027, 48049, 48061, 48069, 48081, 48091, 48099, 48103, 48115, 48117, 48133, 48135, 48157, 48177, 48187, 48195, 48205, 48229, 48231, 48243, 48249, 48259, 48279, 48307, 48343, 48345, 48351, 48355, 48369, 48373, 48375, 48385, 48399, 48415, 48439, 48447, 48451, 48481, 48499, 48501, 48535, 48541, 48559, 48583, 48585, 48615, 48639, 48645, 48657, 48669, 48679, 48681, 48687, 48691, 48703, 48721, 48729, 48741, 48751, 48753, 48777, 48799, 48817, 48831, 48837, 48847, 48849, 48879, 48895, 48897, 48913, 48919, 48921, 48925, 48943, 48955, 48961, 48963, 48967, 48999, 49015, 49021, 49045, 49059, 49081, 49099, 49101, 49111, 49113, 49119, 49123, 49141, 49147, 49149, 49155, 49165, 49203, 49207, 49209, 49213, 49219, 49249, 49275, 49279, 49309, 49315, 49329, 49341, 49359, 49363, 49371, 49375, 49377, 49395, 49399, 49435, 49437, 49447, 49471, 49479, 49489, 49497, 49503, 49507, 49525, 49527, 49561, 49563, 49567, 49585, 49615, 49617, 49623, 49627, 49633, 49635, 49675, 49681, 49689, 49713, 49717, 49719, 49723, 49731, 49743, 49753, 49759, 49779, 49795, 49807, 49819, 49821, 49825, 49833, 49839, 49845, 49855, 49857, 49863, 49869, 49917, 49921, 49927, 49959, 49971, 49981, 50001, 50007, 50023, 50025, 50053, 50059, 50085, 50095, 50101, 50109, 50119, 50139, 50151, 50161, 50169, 50179, 50181, 50203, 50217, 50227, 50247, 50257, 50275, 50287, 50317, 50319, 50323, 50325, 50341, 50367, 50371, 50383, 50407, 50415, 50421, 50431, 50445, 50449, 50463, 50469, 50499, 50517, 50527, 50529, 50551, 50559, 50563, 50575, 50595, 50599, 50625, 50631, 50637, 50643, 50653, 50673, 50677, 50685, 50695, 50697, 50719, 50725, 50739, 50751, 50757, 50763, 50767, 50779, 50845, 50871, 50877, 50887, 50889, 50893, 50905, 50907, 50917, 50929, 50935, 50937, 50983, 50991, 51001, 51009, 51013, 51019, 51039, 51067, 51073, 51085, 51093, 51099, 51105, 51127, 51135, 51139, 51141, 51157, 51159, 51163, 51165, 51169, 51187, 51189, 51207, 51229, 51253, 51255, 51271, 51283, 51289, 51303, 51319, 51325, 51349, 51355, 51379, 51393, 51397, 51399, 51411, 51417, 51421, 51453, 51459, 51475, 51543, 51547, 51549, 51559, 51567, 51571, 51579, 51583, 51597, 51601, 51603, 51609, 51649, 51651, 51673, 51675, 51681, 51703, 51729, 51733, 51747, 51753, 51759, 51777, 51793, 51835, 51837, 51855, 51859, 51861, 51865, 51871, 51903, 51913, 51921, 51937, 51955, 51963, 51979, 52039, 52041, 52045, 52059, 52063, 52071, 52081, 52083, 52089, 52101, 52113, 52149, 52165, 52167, 52179, 52185, 52189, 52219, 52227, 52243, 52249, 52251, 52261, 52275, 52279, 52281, 52291, 52321, 52327, 52339, 52353, 52357, 52365, 52369, 52387, 52425, 52429, 52437, 52441, 52449, 52465, 52467, 52479, 52485, 52491, 52503, 52531, 52545, 52549, 52555, 52567, 52573, 52575, 52611, 52617, 52621, 52627, 52635, 52641, 52659, 52671, 52677, 52683, 52689, 52701, 52719, 52731, 52737, 52753, 52755, 52765, 52785, 52807, 52831, 52845, 52849, 52857, 52861, 52863, 52867, 52879, 52887, 52891, 52903, 52921, 52935, 52951, 52953, 52957, 52963, 52975, 52993, 53005, 53053, 53059, 53067, 53079, 53113, 53145, 53155, 53205, 53235, 53239, 53241, 53245, 53265, 53271, 53287, 53301, 53305, 53307, 53319, 53323, 53329, 53341, 53343, 53361, 53385, 53391, 53407, 53409, 53455, 53457, 53467, 53473, 53475, 53491, 53493, 53499, 53517, 53521, 53533, 53557, 53559, 53563, 53575, 53581, 53593, 53595, 53619, 53623, 53629, 53659, 53697, 53701, 53719, 53745, 53749, 53751, 53773, 53781, 53811, 53815, 53817, 53823, 53827, 53839, 53851, 53865, 53871, 53875, 53881, 53901, 53907, 53941, 53949, 53953, 53971, 53991, 54001, 54007, 54043, 54063, 54069, 54075, 54087, 54103, 54129, 54133, 54141, 54147, 54153, 54163, 54169, 54189, 54223, 54229, 54231, 54247, 54259, 54265, 54273, 54277, 54279, 54285, 54309, 54319, 54327, 54339, 54369, 54373, 54403, 54423, 54445, 54447, 54463, 54465, 54469, 54475, 54481, 54483, 54487, 54517, 54525, 54537, 54543, 54547, 54583, 54589, 54591, 54603, 54621, 54625, 54627, 54637, 54645, 54655, 54673, 54687, 54717, 54721, 54729, 54733, 54735, 54747, 54753, 54777, 54781, 54783, 54801, 54819, 54823, 54841, 54853, 54855, 54879, 54909, 54915, 54921, 54939, 54943, 54949, 54961, 54991, 55005, 55009, 55023, 55029, 55033, 55035, 55045, 55047, 55053, 55063, 55075, 55087, 55093, 55095, 55107, 55129, 55131, 55141, 55159, 55167, 55179, 55191, 55195, 55197, 55219, 55237, 55243, 55255, 55281, 55293, 55303, 55317, 55335, 55351, 55369, 55377, 55387, 55423, 55425, 55477, 55485, 55489, 55503, 55507, 55509, 55515, 55539, 55555, 55569, 55573, 55603, 55609, 55617, 55621, 55633, 55639, 55641, 55665, 55671, 55681, 55693, 55705, 55707, 55713, 55729, 55735, 55741, 55747, 55765, 55801, 55803, 55809, 55819, 55825, 55831, 55863, 55867, 55869, 55885, 55903, 55911, 55929, 55935, 55951, 55953, 55975, 56011, 56029, 56059, 56071, 56077, 56083, 56085, 56091, 56103, 56115, 56119, 56125, 56139, 56181, 56185, 56187, 56203, 56205, 56209, 56251, 56263, 56265, 56269, 56271, 56275, 56283, 56289, 56307, 56311, 56313, 56323, 56347, 56359, 56377, 56385, 56397, 56409, 56437, 56439, 56449, 56461, 56463, 56469, 56479, 56497, 56499, 56517, 56523, 56527, 56533, 56535, 56547, 56563, 56565, 56589, 56637, 56649, 56661, 56673, 56679, 56685, 56703, 56721, 56725, 56731, 56751, 56755, 56775, 56779, 56787, 56797, 56799, 56817, 56839, 56841, 56847, 56869, 56895, 56899, 56931, 56935, 56943, 56955, 56959, 56989, 57021, 57025, 57027, 57045, 57049, 57061, 57081, 57109, 57111, 57115, 57145, 57153, 57217, 57225, 57229, 57241, 57249, 57259, 57267, 57277, 57291, 57301, 57303, 57309, 57313, 57333, 57343, 57345, 57355, 57361, 57397, 57405, 57417, 57429, 57441, 57489, 57499, 57505, 57507, 57523, 57525, 57541, 57561, 57591, 57603, 57607, 57613, 57619, 57625, 57633, 57657, 57667, 57679, 57687, 57691, 57699, 57715, 57717, 57729, 57739, 57757, 57777, 57801, 57805, 57819, 57823, 57825, 57841, 57877, 57879, 57889, 57897, 57907, 57945, 57949, 57951, 57967, 57997, 58003, 58009, 58011, 58023, 58047, 58059, 58075, 58077, 58087, 58117, 58135, 58137, 58159, 58161, 58171, 58183, 58197, 58203, 58227, 58233, 58237, 58243, 58261, 58263, 58287, 58311, 58317, 58323, 58327, 58339, 58351, 58383, 58387, 58389, 58393, 58417, 58423, 58435, 58443, 58447, 58455, 58465, 58467, 58479, 58507, 58509, 58519, 58527, 58549, 58557, 58561, 58605, 58615, 58635, 58639, 58641, 58645, 58663, 58687, 58689, 58695, 58699, 58701, 58719, 58737, 58741, 58753, 58779, 58785, 58795, 58815, 58833, 58845, 58849, 58851, 58867, 58875, 58893, 58897, 58905, 58915, 58969, 58971, 58977, 58989, 58999, 59005, 59019, 59037, 59041, 59043, 59055, 59079, 59083, 59095, 59097, 59125, 59127, 59143, 59145, 59149, 59191, 59233, 59235, 59245, 59257, 59263, 59271, 59283, 59287, 59325, 59329, 59353, 59383, 59391, 59401, 59409, 59413, 59439, 59443, 59445, 59457, 59473, 59479, 59481, 59503, 59517, 59521, 59535, 59547, 59581, 59583, 59587, 59599, 59601, 59605, 59631, 59635, 59649, 59653, 59661, 59665, 59695, 59697, 59707, 59725, 59727, 59737, 59739, 59745, 59761, 59791, 59803, 59823, 59863, 59865, 59871, 59893, 59913, 59917, 59919, 59923, 59943, 59997, 60001, 60007, 60031, 60043, 60045, 60051, 60055, 60061, 60063, 60075, 60085, 60105, 60109, 60117, 60133, 60145, 60147, 60151, 60157, 60175, 60177, 60187, 60217, 60219, 60229, 60231, 60259, 60301, 60307, 60313, 60325, 60337, 60339, 60355, 60369, 60385, 60403, 60433, 60465, 60469, 60483, 60489, 60505, 60513, 60529, 60531, 60535, 60547, 60555, 60559, 60567, 60589, 60595, 60613, 60615, 60619, 60631, 60637, 60639, 60655, 60657, 60675, 60705, 60711, 60745, 60747, 60757, 60763, 60783, 60787, 60801, 60817, 60825, 60843, 60861, 60867, 60889, 60891, 60895, 60909, 60913, 60921, 60925, 60927, 60937, 60955, 60973, 60975, 60993, 60997, 61015, 61017, 61057, 61069, 61077, 61089, 61101, 61119, 61123, 61131, 61153, 61155, 61165, 61173, 61183, 61189, 61197, 61207, 61237, 61243, 61257, 61269, 61299, 61305, 61309, 61329, 61335, 61363, 61377, 61383, 61393, 61399, 61405, 61425, 61437, 61455, 61477, 61503, 61513, 61525, 61551, 61557, 61563, 61581, 61585, 61597, 61615, 61617, 61635, 61663, 61683, 61687, 61699, 61707, 61719, 61723, 61773, 61785, 61791, 61809, 61813, 61825, 61833, 61857, 61875, 61881, 61911, 61915, 61917, 61935, 61953, 61971, 61977, 61999, 62005, 62017, 62025, 62037, 62047, 62067, 62071, 62077, 62079, 62097, 62107, 62109, 62143, 62149, 62151, 62155, 62163, 62167, 62173, 62191, 62193, 62197, 62215, 62227, 62245, 62265, 62287, 62293, 62295, 62299, 62323, 62349, 62359, 62379, 62403, 62421, 62425, 62449, 62467, 62475, 62485, 62499, 62503, 62509, 62517, 62527, 62529, 62533, 62547, 62551, 62583, 62605, 62607, 62613, 62623, 62625, 62629, 62635, 62643, 62659, 62671, 62673, 62701, 62719, 62721, 62733, 62769, 62781, 62797, 62803, 62811, 62815, 62817, 62821, 62863, 62877, 62907, 62911, 62919, 62923, 62929, 62941, 62943, 62961, 62979, 62983, 62985, 62991, 63025, 63033, 63037, 63063, 63069, 63085, 63087, 63111, 63115, 63127, 63133, 63151, 63163, 63171, 63177, 63181, 63193, 63195, 63219, 63223, 63231, 63243, 63255, 63265, 63267, 63283, 63303, 63307, 63319, 63321, 63325, 63339, 63379, 63387, 63409, 63411, 63447, 63457, 63471, 63475, 63483, 63487, 63489, 63495, 63505, 63535, 63547, 63549, 63553, 63583, 63609, 63613, 63637, 63645, 63651, 63661, 63681, 63699, 63703, 63727, 63729, 63741, 63747, 63763, 63765, 63777, 63781, 63789, 63801, 63829, 63835, 63843, 63855, 63865, 63889, 63895, 63903, 63915, 63925, 63945, 63957, 63997, 64015, 64021, 64023, 64039, 64051, 64059, 64071, 64123, 64137, 64147, 64159, 64179, 64185, 64189, 64197, 64203, 64207, 64209, 64213, 64221, 64239, 64251, 64267, 64275, 64327, 64329, 64333, 64345, 64347, 64357, 64369, 64375, 64377, 64387, 64435, 64441, 64449, 64455, 64495, 64497, 64501, 64503, 64521, 64533, 64537, 64545, 64561, 64575, 64579, 64597, 64599, 64609, 64623, 64629, 64647, 64659, 64671, 64705, 64711, 64723, 64785, 64789, 64801, 64813, 64819, 64831, 64833, 64837, 64843, 64851, 64857, 64861, 64881, 64891, 64897, 64903, 64905, 64921, 64923, 64933, 64959, 64963, 64969, 64977, 64995, 65025, 65029, 65037, 65049, 65091, 65101, 65109, 65121, 65125, 65143, 65145, 65163, 65173, 65175, 65187, 65197, 65217, 65235, 65241, 65259, 65281, 65289, 65299, 65341, 65343, 65353, 65367, 65385, 65397, 65427, 65431, 65439, 65457, 65479, 65481, 65493, 65499, 65503, 65521, 65529, 65551, 65553, 65563, 65595, 65599, 65629, 65653, 65661, 65671, 65677, 65683, 65691, 65695, 65713, 65725, 65739, 65745, 65755, 65757, 65779, 65793, 65797, 65803, 65815, 65839, 65857, 65865, 65881, 65883, 65901, 65907, 65913, 65935, 65941, 65977, 65985, 65991, 66003, 66031, 66037, 66039, 66049, 66055, 66057, 66061, 66069, 66073, 66103, 66109, 66123, 66141, 66159, 66163, 66171, 66187, 66193, 66201, 66219, 66235, 66247, 66249, 66255, 66259, 66261, 66267, 66291, 66297, 66325, 66339, 66349, 66355, 66369, 66387, 66427, 66433, 66445, 66447, 66469, 66477, 66489, 66495, 66517, 66529, 66543, 66549, 66553, 66559, 66561, 66565, 66573, 66577, 66579, 66607, 66613, 66637, 66657, 66661, 66667, 66669, 66675, 66699, 66703, 66705, 66723, 66729, 66739, 66747, 66781, 66811, 66823, 66829, 66865, 66873, 66897, 66907, 66915, 66921, 66939, 66943, 66957, 66975, 66985, 67003, 67011, 67015, 67017, 67023, 67057, 67063, 67065, 67075, 67105, 67107, 67117, 67125, 67129, 67131, 67143, 67159, 67189, 67195, 67207, 67209, 67225, 67231, 67243, 67245, 67275, 67285, 67287, 67293, 67329, 67333, 67335, 67339, 67351, 67359, 67417, 67419, 67425, 67443, 67447, 67453, 67473, 67477, 67485, 67503, 67569, 67573, 67579, 67581, 67587, 67591, 67599, 67605, 67641, 67671, 67693, 67695, 67711, 67725, 67731, 67755, 67759, 67791, 67797, 67803, 67837, 67839, 67851, 67861, 67873, 67875, 67887, 67899, 67903, 67915, 67917, 67923, 67927, 67929, 67947, 67963, 67965, 67993, 68025, 68029, 68031, 68043, 68047, 68053, 68061, 68065, 68077, 68085, 68095, 68107, 68113, 68115, 68145, 68181, 68187, 68209, 68217, 68221, 68241, 68253, 68293, 68299, 68313, 68335, 68355, 68365, 68377, 68385, 68397, 68407, 68409, 68425, 68439, 68449, 68461, 68463, 68467, 68469, 68497, 68511, 68527, 68529, 68535, 68547, 68557, 68559, 68577, 68589, 68593, 68595, 68599, 68607, 68643, 68653, 68661, 68671, 68683, 68691, 68755, 68767, 68769, 68779, 68799, 68803, 68809, 68811, 68827, 68829, 68845, 68865, 68869, 68875, 68883, 68895, 68913, 68925, 68929, 68931, 68947, 68965, 68989, 68991, 69009, 69015, 69027, 69033, 69039, 69051, 69073, 69099, 69115, 69117, 69123, 69135, 69141, 69153, 69163, 69165, 69195, 69205, 69219, 69225, 69229, 69237, 69243, 69259, 69271, 69291, 69301, 69309, 69333, 69343, 69367, 69373, 69385, 69393, 69397, 69417, 69435, 69439, 69441, 69451, 69481, 69495, 69511, 69513, 69519, 69523, 69535, 69543, 69565, 69577, 69583, 69595, 69615, 69625, 69627, 69637, 69645, 69649, 69679, 69681, 69687, 69703, 69709, 69723, 69727, 69729, 69747, 69757, 69775, 69787, 69789, 69793, 69817, 69819, 69825, 69841, 69871, 69891, 69919, 69921, 69939, 69951, 69985, 69999, 70035, 70063, 70071, 70077, 70081, 70089, 70105, 70107, 70125, 70131, 70165, 70173, 70207, 70215, 70219, 70227, 70261, 70267, 70269, 70287, 70299, 70317, 70333, 70351, 70353, 70357, 70359, 70371, 70405, 70417, 70423, 70435, 70447, 70449, 70455, 70459, 70465, 70471, 70485, 70501, 70509, 70519, 70539, 70545, 70573, 70591, 70597, 70623, 70635, 70639, 70657, 70665, 70669, 70671, 70677, 70699, 70701, 70707, 70717, 70741, 70755, 70761, 70765, 70773, 70783, 70791, 70795, 70803, 70813, 70815, 70845, 70849, 70857, 70867, 70875, 70899, 70923, 70945, 70983, 71005, 71017, 71025, 71037, 71047, 71049, 71055, 71073, 71089, 71101, 71115, 71157, 71161, 71169, 71173, 71179, 71191, 71199, 71215, 71221, 71223, 71235, 71239, 71275, 71277, 71283, 71305, 71329, 71341, 71347, 71359, 71371, 71379, 71403, 71421, 71431, 71433, 71445, 71455, 71479, 71493, 71517, 71535, 71539, 71547, 71557, 71569, 71577, 71593, 71599, 71601, 71617, 71619, 71623, 71631, 71641, 71673, 71679, 71697, 71725, 71739, 71743, 71749, 71769, 71793, 71803, 71821, 71829, 71833, 71869, 71871, 71899, 71929, 71931, 71949, 71953, 71959, 71989, 71991, 71997, 72001, 72019, 72021, 72025, 72055, 72061, 72075, 72123, 72127, 72135, 72139, 72141, 72151, 72157, 72171, 72183, 72187, 72211, 72223, 72235, 72243, 72249, 72265, 72273, 72277, 72303, 72313, 72327, 72333, 72379, 72391, 72393, 72403, 72411, 72439, 72459, 72463, 72493, 72495, 72499, 72501, 72519, 72535, 72565, 72577, 72585, 72589, 72591, 72601, 72607, 72609, 72649, 72663, 72669, 72687, 72693, 72715, 72727, 72735, 72751, 72753, 72765, 72787, 72799, 72819, 72865, 72891, 72895, 72897, 72903, 72907, 72921, 72939, 72955, 72975, 72979, 72985, 72991, 72997, 73003, 73009, 73017, 73021, 73023, 73053, 73065, 73081, 73087, 73101, 73113, 73117, 73131, 73149, 73155, 73165, 73177, 73179, 73185, 73197, 73213, 73221, 73227, 73231, 73249, 73273, 73291, 73305, 73335, 73341, 73363, 73377, 73387, 73405, 73417, 73419, 73431, 73441, 73449, 73461, 73465, 73467, 73473, 73495, 73537, 73543, 73569, 73585, 73587, 73617, 73627, 73629, 73633, 73639, 73669, 73671, 73677, 73683, 73717, 73723, 73731, 73735, 73777, 73785, 73809, 73821, 73843, 73851, 73873, 73899, 73911, 73915, 73933, 73935, 73969, 73977, 73995, 74013, 74031, 74035, 74041, 74049, 74059, 74071, 74101, 74103, 74109, 74121, 74139, 74143, 74155, 74161, 74175, 74187, 74193, 74197, 74199, 74221, 74227, 74257, 74259, 74289, 74293, 74307, 74311, 74325, 74329, 74331, 74347, 74349, 74353, 74355, 74385, 74391, 74415, 74425, 74433, 74439, 74445, 74449, 74455, 74457, 74487, 74497, 74503, 74509, 74515, 74541, 74559, 74571, 74577, 74581, 74583, 74595, 74613, 74635, 74643, 74647, 74697, 74701, 74721, 74733, 74739, 74749, 74761, 74767, 74773, 74787, 74793, 74797, 74815, 74817, 74857, 74859, 74865, 74869, 74893, 74895, 74907, 74911, 74913, 74919, 74941, 74955, 74959, 74973, 75019, 75025, 75055, 75067, 75099, 75103, 75111, 75117, 75147, 75163, 75181, 75193, 75195, 75201, 75211, 75213, 75223, 75229, 75247, 75255, 75265, 75271, 75289, 75297, 75307, 75319, 75357, 75391, 75399, 75421, 75433, 75435, 75441, 75475, 75481, 75483, 75505, 75511, 75517, 75525, 75529, 75537, 75549, 75561, 75567, 75583, 75585, 75589, 75591, 75603, 75625, 75631, 75637, 75645, 75655, 75675, 75679, 75693, 75709, 75717, 75733, 75735, 75741, 75759, 75781, 75789, 75793, 75799, 75823, 75825, 75835, 75841, 75853, 75859, 75867, 75873, 75885, 75895, 75901, 75925, 75937, 75945, 75967, 75985, 75991, 75993, 76015, 76029, 76045, 76065, 76077, 76083, 76089, 76105, 76107, 76113, 76125, 76129, 76149, 76179, 76183, 76189, 76203, 76209, 76219, 76233, 76239, 76275, 76303, 76317, 76339, 76341, 76357, 76359, 76371, 76381, 76393, 76405, 76423, 76455, 76461, 76473, 76497, 76503, 76525, 76533, 76551, 76555, 76567, 76587, 76609, 76615, 76617, 76623, 76633, 76639, 76663, 76671, 76705, 76713, 76717, 76735, 76737, 76749, 76759, 76779, 76783, 76785, 76797, 76801, 76803, 76831, 76849, 76851, 76873, 76885, 76893, 76905, 76939, 76947, 76969, 76975, 76989, 76995, 77001, 77017, 77037, 77053, 77061, 77065, 77091, 77115, 77127, 77137, 77149, 77167, 77209, 77211, 77217, 77223, 77245, 77253, 77263, 77269, 77275, 77281, 77289, 77293, 77301, 77307, 77353, 77355, 77365, 77377, 77379, 77395, 77397, 77407, 77431, 77457, 77473, 77481, 77503, 77505, 77515, 77521, 77527, 77533, 77553, 77565, 77577, 77599, 77601, 77617, 77619, 77625, 77653, 77661, 77671, 77679, 77689, 77721, 77733, 77745, 77757, 77763, 77767, 77785, 77787, 77791, 77805, 77811, 77821, 77835, 77841, 77853, 77857, 77869, 77877, 77893, 77911, 77913, 77937, 77955, 77961, 77983, 78015, 78025, 78067, 78069, 78079, 78093, 78099, 78109, 78129, 78133, 78141, 78145, 78165, 78187, 78193, 78195, 78199, 78253, 78261, 78271, 78277, 78279, 78283, 78291, 78319, 78321, 78325, 78339, 78355, 78357, 78373, 78379, 78385, 78397, 78409, 78469, 78487, 78511, 78529, 78531, 78553, 78591, 78597, 78603, 78613, 78627, 78645, 78649, 78657, 78661, 78673, 78675, 78679, 78711, 78723, 78729, 78733, 78739, 78751, 78763, 78765, 78775, 78825, 78829, 78837, 78849, 78855, 78861, 78867, 78879, 78901, 78913, 78919, 78945, 78969, 78973, 78985, 78987, 78991, 79003, 79033, 79045, 79057, 79099, 79113, 79117, 79129, 79135, 79137, 79153, 79159, 79161, 79177, 79183, 79207, 79215, 79221, 79255, 79261, 79269, 79275, 79327, 79341, 79353, 79365, 79369, 79371, 79381, 79401, 79413, 79423, 79429, 79449, 79453, 79465, 79467, 79473, 79477, 79491, 79497, 79509, 79513, 79531, 79537, 79551, 79557, 79561, 79569, 79593, 79615, 79623, 79657, 79663, 79669, 79675, 79683, 79701, 79705, 79707, 79725, 79737, 79761, 79773, 79825, 79851, 79855, 79885, 79891, 79899, 79905, 79915, 79917, 79921, 79933, 79959, 79981, 79983, 79989, 79993, 80001, 80023, 80041, 80047, 80059, 80073, 80077, 80107, 80137, 80157, 80179, 80185, 80199, 80203, 80209, 80211, 80241, 80269, 80277, 80287, 80295, 80305, 80311, 80313, 80317, 80335, 80337, 80341, 80347, 80377, 80395, 80413, 80425, 80433, 80437, 80443, 80473, 80475, 80481, 80503, 80527, 80545, 80557, 80559, 80569, 80577, 80581, 80583, 80593, 80625, 80641, 80649, 80655, 80671, 80683, 80685, 80703, 80733, 80737, 80749, 80767, 80773, 80775, 80797, 80815, 80835, 80839, 80851, 80853, 80875, 80881, 80901, 80907, 80913, 80925, 80937, 80941, 80955, 80967, 81001, 81007, 81009, 81019, 81039, 81081, 81093, 81103, 81111, 81115, 81133, 81145, 81147, 81157, 81165, 81169, 81195, 81207, 81213, 81217, 81241, 81261, 81273, 81277, 81295, 81301, 81307, 81313, 81319, 81337, 81349, 81355, 81357, 81363, 81403, 81421, 81439, 81451, 81459, 81465, 81469, 81475, 81481, 81511, 81525, 81529, 81537, 81553, 81573, 81585, 81601, 81609, 81615, 81621, 81649, 81655, 81661, 81711, 81715, 81721, 81723, 81727, 81733, 81745, 81747, 81757, 81781, 81783, 81811, 81817, 81841, 81849, 81861, 81871, 81883, 81913, 81931, 81933, 81937, 81943, 81949, 81969, 81985, 82035, 82039, 82057, 82059, 82063, 82095, 82101, 82105, 82111, 82119, 82123, 82135, 82155, 82165, 82177, 82185, 82189, 82197, 82227, 82231, 82237, 82257, 82269, 82285, 82293, 82299, 82311, 82321, 82345, 82363, 82383, 82393, 82419, 82429, 82437, 82447, 82449, 82473, 82479, 82501, 82503, 82513, 82519, 82543, 82567, 82581, 82585, 82593, 82635, 82677, 82681, 82687, 82689, 82693, 82705, 82707, 82719, 82731, 82743, 82749, 82761, 82771, 82785, 82791, 82795, 82797, 82803, 82825, 82831, 82833, 82855, 82857, 82861, 82879, 82881, 82893, 82909, 82939, 82957, 82959, 82977, 82983, 82995, 83025, 83035, 83055, 83059, 83067, 83077, 83083, 83097, 83133, 83161, 83167, 83181, 83191, 83209, 83223, 83233, 83271, 83277, 83293, 83307, 83317, 83319, 83337, 83355, 83391, 83401, 83419, 83421, 83443, 83445, 83449, 83455, 83479, 83499, 83521, 83523, 83559, 83563, 83569, 83571, 83593, 83601, 83605, 83611, 83625, 83631, 83643, 83647, 83655, 83667, 83671, 83697, 83707, 83709, 83719, 83733, 83751, 83763, 83769, 83773, 83799, 83815, 83841, 83857, 83865, 83887, 83895, 83901, 83905, 83907, 83917, 83919, 83923, 83929, 83953, 83959, 83961, 83985, 83995, 84009, 84049, 84075, 84085, 84091, 84097, 84105, 84115, 84121, 84135, 84141, 84153, 84157, 84189, 84201, 84213, 84223, 84235, 84241, 84273, 84309, 84319, 84331, 84345, 84381, 84393, 84405, 84411, 84421, 84423, 84429, 84435, 84457, 84463, 84465, 84471, 84483, 84487, 84495, 84499, 84519, 84549, 84553, 84591, 84619, 84631, 84633, 84645, 84661, 84663, 84673, 84675, 84679, 84687, 84693, 84721, 84745, 84769, 84781, 84787, 84789, 84813, 84819, 84829, 84841, 84847, 84849, 84861, 84883, 84895, 84925, 84937, 84939, 84949, 84955, 84957, 84975, 84991, 84999, 85003, 85009, 85033, 85035, 85053, 85075, 85087, 85095, 85105, 85129, 85143, 85147, 85149, 85159, 85177, 85179, 85197, 85239, 85245, 85249, 85255, 85275, 85303, 85305, 85327, 85333, 85335, 85339, 85351, 85353, 85371, 85387, 85419, 85441, 85449, 85461, 85465, 85491, 85495, 85507, 85555, 85563, 85569, 85587, 85591, 85597, 85605, 85617, 85621, 85669, 85671, 85693, 85701, 85705, 85735, 85755, 85773, 85777, 85779, 85807, 85813, 85815, 85819, 85821, 85843, 85851, 85861, 85869, 85873, 85875, 85917, 85923, 85933, 85947, 85963, 85965, 85999, 86005, 86025, 86031, 86037, 86071, 86089, 86095, 86101, 86103, 86125, 86155, 86163, 86167, 86185, 86187, 86199, 86205, 86221, 86227, 86257, 86259, 86269, 86281, 86293, 86323, 86331, 86335, 86341, 86385, 86409, 86415, 86421, 86425, 86427, 86439, 86445, 86469, 86481, 86491, 86505, 86515, 86545, 86571, 86593, 86595, 86617, 86631, 86637, 86647, 86661, 86671, 86677, 86679, 86719, 86721, 86743, 86761, 86763, 86775, 86803, 86815, 86829, 86857, 86865, 86877, 86901, 86925, 86929, 86941, 86949, 86953, 86965, 86973, 86983, 86989, 87003, 87007, 87015, 87019, 87025, 87027, 87049, 87051, 87055, 87097, 87103, 87109, 87111, 87121, 87129, 87133, 87151, 87153, 87171, 87175, 87183, 87199, 87207, 87241, 87255, 87261, 87289, 87297, 87301, 87303, 87319, 87339, 87343, 87361, 87387, 87391, 87397, 87403, 87411, 87415, 87433, 87435, 87451, 87453, 87459, 87469, 87481, 87493, 87499, 87529, 87549, 87559, 87571, 87579, 87583, 87595, 87607, 87633, 87639, 87645, 87667, 87669, 87703, 87705, 87717, 87721, 87733, 87741, 87747, 87751, 87783, 87793, 87795, 87823, 87835, 87837, 87867, 87871, 87873, 87897, 87931, 87937, 87957, 87973, 87979, 87981, 87991, 88011, 88027, 88041, 88045, 88057, 88065, 88075, 88083, 88089, 88117, 88119, 88125, 88129, 88147, 88159, 88171, 88173, 88183, 88185, 88221, 88237, 88249, 88267, 88299, 88305, 88329, 88333, 88335, 88347, 88363, 88371, 88401, 88413, 88423, 88435, 88441, 88459, 88467, 88495, 88503, 88519, 88521, 88527, 88537, 88539, 88551, 88561, 88587, 88591, 88593, 88603, 88615, 88629, 88653, 88663, 88665, 88693, 88707, 88713, 88735, 88747, 88755, 88773, 88783, 88789, 88813, 88815, 88819, 88821, 88837, 88839, 88851, 88861, 88881, 88893, 88897, 88903, 88915, 88927, 88939, 88945, 88947, 88971, 88987, 88989, 89001, 89023, 89025, 89029, 89041, 89043, 89065, 89071, 89085, 89115, 89119, 89131, 89167, 89169, 89175, 89191, 89209, 89217, 89223, 89233, 89257, 89283, 89287, 89293, 89319, 89325, 89335, 89337, 89349, 89355, 89359, 89377, 89383, 89385, 89397, 89403, 89421, 89427, 89433, 89445, 89467, 89481, 89493, 89497, 89503, 89509, 89527, 89533, 89545, 89557, 89575, 89577, 89587, 89599, 89637, 89653, 89695, 89715, 89743, 89755, 89781, 89787, 89799, 89805, 89829, 89841, 89845, 89869, 89871, 89883, 89901, 89905, 89911, 89913, 89923, 89947, 89977, 89989, 90009, 90039, 90063, 90069, 90093, 90097, 90099, 90105, 90121, 90127, 90133, 90159, 90175, 90183, 90199, 90229, 90231, 90241, 90259, 90261, 90265, 90279, 90291, 90295, 90301, 90313, 90349, 90351, 90355, 90357, 90373, 90379, 90393, 90409, 90435, 90457, 90489, 90517, 90535, 90537, 90547, 90561, 90579, 90601, 90603, 90619, 90625, 90649, 90663, 90679, 90691, 90703, 90711, 90727, 90745, 90757, 90765, 90769, 90771, 90813, 90817, 90819, 90825, 90847, 90861, 90871, 90891, 90897, 90913, 90919, 90921, 90939, 90951, 90961, 90963, 90973, 90993, 91015, 91021, 91023, 91039, 91047, 91051, 91059, 91087, 91089, 91111, 91113, 91129, 91131, 91141, 91143, 91177, 91191, 91195, 91213, 91225, 91227, 91237, 91239, 91257, 91269, 91279, 91299, 91303, 91329, 91333, 91339, 91387, 91399, 91405, 91423, 91429, 91447, 91479, 91483, 91491, 91497, 91509, 91527, 91531, 91543, 91561, 91563, 91575, 91581, 91585, 91587, 91593, 91605, 91633, 91639, 91651, 91665, 91701, 91711, 91729, 91731, 91761, 91779, 91783, 91795, 91813, 91815, 91833, 91837, 91843, 91861, 91867, 91885, 91891, 91903, 91917, 91921, 91947, 91965, 91995, 92011, 92043, 92053, 92059, 92073, 92077, 92085, 92095, 92113, 92143, 92155, 92179, 92199, 92205, 92211, 92223, 92233, 92239, 92247, 92275, 92277, 92299, 92301, 92341, 92347, 92367, 92373, 92379, 92395, 92407, 92409, 92425, 92427, 92433, 92443, 92445, 92457, 92521, 92527, 92557, 92559, 92569, 92571, 92589, 92599, 92635, 92647, 92659, 92665, 92677, 92685, 92689, 92697, 92709, 92727, 92739, 92743, 92749, 92757, 92761, 92779, 92781, 92785, 92823, 92829, 92841, 92851, 92869, 92887, 92905, 92911, 92913, 92917, 92925, 92929, 92937, 92941, 92973, 92977, 93003, 93021, 93031, 93057, 93063, 93093, 93099, 93105, 93117, 93123, 93129, 93135, 93139, 93147, 93169, 93189, 93223, 93229, 93241, 93247, 93261, 93265, 93285, 93289, 93295, 93303, 93339, 93355, 93381, 93397, 93403, 93417, 93433, 93451, 93459, 93465, 93483, 93501, 93505, 93507, 93517, 93523, 93525, 93543, 93547, 93555, 93577, 93591, 93597, 93609, 93627, 93633, 93669, 93691, 93703, 93705, 93715, 93753, 93759, 93781, 93787, 93799, 93813, 93829, 93831, 93849, 93871, 93877, 93883, 93885, 93895, 93913, 93915, 93945, 93957, 93963, 93969, 93979, 93987, 94003, 94005, 94009, 94017, 94021, 94029, 94039, 94065, 94069, 94089, 94093, 94105, 94135, 94159, 94189, 94195, 94197, 94209, 94215, 94219, 94221, 94255, 94257, 94273, 94291, 94293, 94299, 94317, 94321, 94323, 94327, 94345, 94353, 94363, 94381, 94389, 94407, 94419, 94423, 94443, 94453, 94461, 94471, 94473, 94479, 94489, 94513, 94531, 94551, 94555, 94567, 94573, 94575, 94593, 94605, 94629, 94641, 94651, 94659, 94663, 94677, 94701, 94711, 94735, 94761, 94767, 94785, 94789, 94797, 94801, 94803, 94821, 94831, 94849, 94857, 94867, 94879, 94891, 94909, 94923, 94933, 94957, 94963, 94965, 94971, 94987, 95025, 95029, 95035, 95037, 95055, 95059, 95089, 95113, 95143, 95145, 95163, 95179, 95185, 95209, 95223, 95229, 95239, 95245, 95259, 95271, 95281, 95301, 95335, 95343, 95373, 95383, 95385, 95403, 95413, 95415, 95431, 95445, 95449, 95451, 95461, 95467, 95469, 95479, 95497, 95509, 95517, 95539, 95557, 95575, 95577, 95593, 95601, 95607, 95619, 95635, 95637, 95671, 95677, 95683, 95697, 95707, 95731, 95739, 95745, 95751, 95761, 95791, 95793, 95803, 95809, 95811, 95823, 95853, 95859, 95865, 95875, 95877, 95887, 95923, 95929, 95935, 95955, 95995, 96001, 96003, 96013, 96015, 96019, 96027, 96037, 96043, 96055, 96079, 96099, 96129, 96139, 96141, 96147, 96151, 96181, 96187, 96201, 96211, 96231, 96235, 96243, 96267, 96273, 96309, 96313, 96315, 96333, 96343, 96363, 96375, 96391, 96393, 96399, 96403, 96411, 96423, 96427, 96459, 96469, 96475, 96495, 96529, 96549, 96553, 96571, 96579, 96589, 96591, 96601, 96621, 96625, 96631, 96649, 96657, 96663, 96675, 96709, 96715, 96747, 96751, 96769, 96777, 96783, 96813, 96817, 96823, 96835, 96843, 96847, 96861, 96867, 96877, 96879, 96909, 96957, 96963, 96973, 96979, 96981, 97005, 97009, 97053, 97063, 97069, 97083, 97089, 97093, 97105, 97107, 97113, 97213, 97219, 97231, 97233, 97243, 97255, 97261, 97275, 97279, 97323, 97335, 97347, 97359, 97365, 97381, 97387, 97389, 97411, 97413, 97419, 97423, 97443, 97449, 97465, 97467, 97471, 97491, 97495, 97525, 97531, 97545, 97555, 97561, 97575, 97587, 97599, 97603, 97617, 97623, 97629, 97639, 97671, 97681, 97683, 97687, 97699, 97701, 97729, 97743, 97755, 97779, 97807, 97809, 97819, 97821, 97849, 97861, 97875, 97891, 97911, 97917, 97927, 97947, 97957, 97971, 97975, 97987, 97999, 98007, 98041, 98049, 98053, 98065, 98091, 98107, 98127, 98137, 98155, 98161, 98167, 98179, 98185, 98187, 98199, 98217, 98253, 98283, 98289, 98293, 98305, 98313, 98317, 98329, 98349, 98353, 98391, 98395, 98413, 98443, 98451, 98455, 98469, 98473, 98481, 98491, 98511, 98547, 98553, 98563, 98569, 98583, 98587, 98599, 98611, 98619, 98625, 98629, 98637, 98661, 98679, 98689, 98701, 98707, 98709, 98713, 98731, 98755, 98757, 98763, 98799, 98805, 98817, 98827, 98869, 98871, 98877, 98893, 98895, 98935, 98947, 98953, 98979, 98997, 99003, 99015, 99045, 99057, 99069, 99087, 99103, 99109, 99121, 99141, 99147, 99151, 99153, 99165, 99171, 99175, 99207, 99211, 99217, 99225, 99235, 99267, 99271, 99279, 99289, 99301, 99319, 99333, 99337, 99367, 99375, 99381, 99403, 99417, 99459, 99477, 99483, 99493, 99507, 99525, 99531, 99541, 99543, 99547, 99555, 99571, 99573, 99577, 99583, 99585, 99589, 99591, 99595, 99603, 99637, 99649, 99657, 99667, 99673, 99675, 99691, 99703, 99717, 99735, 99745, 99753, 99781, 99795, 99813, 99817, 99823, 99825, 99843, 99855, 99859, 99865, 99877, 99879, 99885, 99889, 99897, 99907, 99919, 99921, 99949, 99955, 99973, 99987};

int find(int n) {

int l = 0, r = 8772;

while (r - l > 1) {

int mid = (l+r) / 2;

if (a[mid] > n) {

r = mid;

} else {

l = mid;

}

}

return l;

}

int main() {

int n, m;

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

printf("%d\n", find(m) - find(n));

}

**C**

#include<stdio.h>

#define MAXN 1000010

int flag[MAXN];

int m,n,a[MAXN],s[MAXN],size=0;

int fa(int k)

{

if(flag[k])

return a[k];

return fa(k-1);

}

int main()

{

int i, p, k, j;

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

for(i=1;i<=n;i+=2)

{

s[++size]=i;

flag[i]=1;

a[i]=size;

}

for(i=2;i<=size;i++)

{

int Mod=s[i],d=s[i]-1;

if(Mod>size)

break;

for(p=1,j=Mod;j<=size;j+=Mod,p++)

{

flag[s[j]]=0;

for(k=1;k<Mod&&k+j<=size;k++)

{

s[++d]=s[j+k];

a[s[j+k]]-=p;

}

}

size=d;

}

printf("%d\n",fa(n-1)-fa(m));

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.StringTokenizer;

class Reader11{

static BufferedReader reader;

static StringTokenizer tokenizer;

static void init(InputStream input){

reader=new BufferedReader(new InputStreamReader(input));

tokenizer=new StringTokenizer("");

}

static String next() throws IOException{

while (!tokenizer.hasMoreElements()) {

tokenizer =new StringTokenizer(reader.readLine());

}

return tokenizer.nextToken();

}

static int nextInt() throws IOException{

return Integer.parseInt(next());

}

}

public class Main{

/\*\*

\* @param args

\* @throws IOException

\*/

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

// TODO Auto-generated method stub

Reader11.init(System.in);

int m=Reader11.nextInt();

int n=Reader11.nextInt();

ArrayList<Integer> a=new ArrayList<Integer>();

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

a.add(i);

}

int k=1;

int t=a.get(k);

while (t<=a.size()) {

int t0=a.get(k+1);

for (int j = 1; t\*j < a.size()+j; j++) {

a.remove(t\*j-j);

}

if (a.get(k)==t0) {

}else {

k++;

}

t=a.get(k);

}

int Mj = 0;

for (int i = 0; i < a.size(); i++) {

if (a.get(i)>m) {

Mj=i;

break;

}else if(a.get(i)==m){

Mj=i+1;

break;

}

}

System.out.println(a.size()-Mj);

}

}

PREV-9**历届试题 大臣的旅费**

时间限制：1.0s   内存限制：256.0MB

关键字 深度优先遍历

问题描述

很久以前，T王国空前繁荣。为了更好地管理国家，王国修建了大量的快速路，用于连接首都和王国内的各大城市。

为节省经费，T国的大臣们经过思考，制定了一套优秀的修建方案，使得任何一个大城市都能从首都直接或者通过其他大城市间接到达。同时，如果不重复经过大城市，从首都到达每个大城市的方案都是唯一的。

J是T国重要大臣，他巡查于各大城市之间，体察民情。所以，从一个城市马不停蹄地到另一个城市成了J最常做的事情。他有一个钱袋，用于存放往来城市间的路费。

聪明的J发现，如果不在某个城市停下来修整，在连续行进过程中，他所花的路费与他已走过的距离有关，在走第x千米到第x+1千米这一千米中（x是整数），他花费的路费是x+10这么多。也就是说走1千米花费11，走2千米要花费23。

J大臣想知道：他从某一个城市出发，中间不休息，到达另一个城市，所有可能花费的路费中最多是多少呢？

输入格式

输入的第一行包含一个整数n，表示包括首都在内的T王国的城市数

城市从1开始依次编号，1号城市为首都。

接下来n-1行，描述T国的高速路（T国的高速路一定是n-1条）

每行三个整数Pi, Qi, Di，表示城市Pi和城市Qi之间有一条高速路，长度为Di千米。

输出格式

输出一个整数，表示大臣J最多花费的路费是多少。

样例输入1

5  
1 2 2  
1 3 1  
2 4 5  
2 5 4

样例输出1

135

输出格式

大臣J从城市4到城市5要花费135的路费。

**参考代码**

**C++**

#include <iostream>

#include <vector>

#include <cstdio>

using namespace std;

struct edge

{

int Number;

int Length;

edge\* next;

edge(int number, int length)

{

Number = number;

Length = length;

next = NULL;

}

};

struct vert

{

int Number;

int PathLen;

bool Isvisit;

edge\* First;

vert(int number)

{

Number = number;

Isvisit = false;

First = NULL;

PathLen = 0;

}

};

void dfs(vector<vert>& GList, int num)

{

edge\* p= GList[num].First;

GList[num].Isvisit = true;

for (; p!=NULL; p=p->next)

{

if(GList[p->Number].Isvisit == false)

{

GList[p->Number].PathLen =GList[num].PathLen + p->Length;

dfs(GList, p->Number);

}

}

}

int main()

{

vector<vert> GList;

int n, i, j;

int fir, sed, len;

cin>>n;

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

{

GList.push\_back(vert(i));

}

for(j = 1; j < n; j++)

{

edge\* p1, \*p2;

scanf("%d%d%d",&fir,&sed,&len);

p1 = GList[fir].First;

p2 = GList[sed].First;

//无向图

if (p1==NULL)

{

GList[fir].First = new edge(sed, len);

}

else

{

while(p1->next!=NULL) //将边链接到顶点

{

p1 = p1->next;

}

p1->next = new edge(sed, len);

}

if (p2==NULL)

{

GList[sed].First = new edge(fir, len);

}

else

{

while(p2->next!=NULL) //将边链接到顶点

{

p2 = p2->next;

}

p2->next = new edge(fir, len);

}

}

int start, end, length = 0;

dfs(GList, 1);

for (j = 1; j<=n; j++)

{

if (GList[j].PathLen > length)//更新最长距离

{

start = i;

end = j;

length = GList[j].PathLen;

}

GList[j].PathLen = 0;

GList[j].Isvisit = false;

}

dfs(GList, end);

for (j = 1; j<=n; j++)

{

if (GList[j].PathLen > length)//更新最长距离

{

start = i;

end = j;

length = GList[j].PathLen;

}

GList[j].PathLen = 0;

GList[j].Isvisit = false;

}

int cost = length\*10;

for (i = 1; i <= length; i++)

{

cost+=i;

}

cout<<cost<<endl;

return 0;

}

**C**

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

struct node;

typedef struct node Node;

typedef Node \*PtrToNode;

typedef PtrToNode List;

typedef PtrToNode Position;

struct node

{

int n;

int val;

Position next;

};

int count=0;

int max=0;

int x;

int \*visit; //是否已遍历

Position Last(List l); //找出最后项

void Insert(int x,int q,List l,Position p); //在p后插入含x的项

void Dfs(int a,List l[]); //深度优先搜索

int Num(List l);

int main(void)

{

int n,u,v,q,a,b;

int i,j,k;

Node \*head;

List \*l,tmp;

Position p;

fscanf(stdin,"%d",&n);

head=(Node \*)malloc(sizeof(Node)\*(n+1));

l=(List \*)malloc(sizeof(List)\*(n+1));

visit=(int \*)malloc(sizeof(int)\*(n+1));

for(i=0;i<=n;i++) //初始化表头及链表

{

head[i].next=NULL;

l[i]=&head[i];

}

for(i=1;i<=n-1;i++) //建立无向图

{

fscanf(stdin,"%d%d%d",&u,&v,&q);

Insert(v,q,l[u],Last(l[u]));

Insert(u,q,l[v],Last(l[v]));

}

for(j=1;j<=n;j++)

visit[j]=0;

Dfs(1,l); //第一次遍历，找到点a，用全局变量x保存

for(j=1;j<=n;j++)

visit[j]=0;

count=0;

max=0;

Dfs(x,l); //第二次遍历，找到点b，用全局变量x保存，此时max为最大距离

printf("%d",max\*10+(max+1)\*max/2);

return 0;

}

Position Last(List l)

{

Position p;

for(p=l;p->next!=NULL;p=p->next);

return p;

}

void Insert(int x,int q,List l,Position p)

{

Position tmp;

tmp=(Position) malloc(sizeof(Node));

tmp->n=x;

tmp->val=q;

tmp->next=p->next;

p->next=tmp;

}

void Dfs(int a,List l[])

{

Position p;

visit[a]=1;

for(p=l[a]->next;p!=NULL;p=p->next)

if(!(visit[p->n]))

{

count+=p->val;

if(count>max)

{

max=count;

x=p->n;

}

Dfs(p->n,l);

count-=p->val;

}

}

int Num(List l)

{

int n=0;

Position p;

for(p=l->next;p!=NULL;p=p->next)

n++;

return n;

}

**JAVA**

import java.io.BufferedInputStream;

import java.io.IOException;

import java.util.ArrayList;

public class Main

{

private static BufferedInputStream in = new BufferedInputStream(System.in);

private static ArrayList<Node> n = new ArrayList<Node>();

private static Integer dis = 0;

private static Integer pow = 0;

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

{

int size = readInt();

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

{

n.add(null);

}

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

{

int x = readInt()-1;

int y = readInt()-1;

int d = readInt();

Node node = new Node();

node.nextSide = n.get(x);

node.con = y;

node.power = d;

n.set(x, node);

node = new Node();

node.nextSide = n.get(y);

node.con = x;

node.power = d;

n.set(y, node);

}

away(0, 0, -1);

pow = 0;

away(dis, 0, -1);

System.out.println(pow \* 10 + (1 + pow) \* pow /2);

}

private static void away(int index, int power, int from)

{

if(pow < power)

{

pow = power;

dis = index;

}

Node node = n.get(index);

while(node != null)

{

if(node.con == from)

{

node = node.nextSide;

continue;

}

away(node.con, power + node.power, index);

node = node.nextSide;

}

}

private static int readInt() throws IOException

{

int i,sum=0;

while(((i=in.read())&48) != 48 || i>57);

for(;(i&56) == 48 || (i&62) == 56; i=in.read())

sum = sum\*10 + (i&15);

return sum;

}

private static class Node

{

int power;

int con;

Node nextSide;

}

}

PREV-8**历届试题 买不到的数目**

时间限制：1.0s   内存限制：256.0MB

关键字 数论 动态规划

问题描述

小明开了一家糖果店。他别出心裁：把水果糖包成4颗一包和7颗一包的两种。糖果不能拆包卖。

小朋友来买糖的时候，他就用这两种包装来组合。当然有些糖果数目是无法组合出来的，比如要买 10 颗糖。

你可以用计算机测试一下，在这种包装情况下，最大不能买到的数量是17。大于17的任何数字都可以用4和7组合出来。

本题的要求就是在已知两个包装的数量时，求最大不能组合出的数字。

输入格式

两个正整数，表示每种包装中糖的颗数(都不多于1000)

输出格式

一个正整数，表示最大不能买到的糖数

样例输入1

4 7

样例输出1

17

样例输入2

3 5

样例输出2

7

**参考代码**

**C++**

#include<iostream>

using namespace std;

int main()

{

int a,b;

cin>>a>>b;

cout<<a\*b-a-b<<endl;

return 0;

}

**C**

#include<stdio.h>

#define MAXSIZE 1000

int main() {

int flag[MAXSIZE] = {0};

int maxunuse[MAXSIZE] = {0};

int num1, num2, minNum, maxNum, temp, i, count = 0;

scanf("%d%d", &num1, &num2);

minNum = ((num1 < num2) ? num1 : num2);

maxNum = num1 + num2 - minNum;

for(i = 1; ;i++){

temp = i \* maxNum % minNum;

if((temp) && flag[temp] == 0){

flag[temp] = 1;

maxunuse[temp] = i \* maxNum -minNum;

count++;

if(count == minNum - 1){

break;

}

}

}

printf("%d\n", maxunuse[temp]);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.InputStreamReader;

public class Main {

public static void main(String[] args) throws Throwable {

BufferedReader buf = new BufferedReader(

new InputStreamReader(System.in));

String strNum = buf.readLine();

String[] num = strNum.split(" ");

buf.close();

int a, b;

a = Integer.valueOf(num[0]);

b = Integer.valueOf(num[1]);

if (a > b) {

int tem = a;

a = b;

b = tem;

}

int c = a \* b;

int tem = c;

while (tem > 0) {

if (tem % a == 0)

tem = --c;

else if (tem % b == 0)

tem = --c;

else

tem -= b;

}

System.out.println(c);

}

}

PREV-7**历届试题 连号区间数**

时间限制：1.0s   内存限制：256.0MB

关键字 并查集

问题描述

小明这些天一直在思考这样一个奇怪而有趣的问题：

在1~N的某个全排列中有多少个连号区间呢？这里所说的连号区间的定义是：

如果区间[L, R] 里的所有元素（即此排列的第L个到第R个元素）递增排序后能得到一个长度为R-L+1的“连续”数列，则称这个区间连号区间。

当N很小的时候，小明可以很快地算出答案，但是当N变大的时候，问题就不是那么简单了，现在小明需要你的帮助。

输入格式

第一行是一个正整数N (1 <= N <= 50000), 表示全排列的规模。

第二行是N个不同的数字Pi(1 <= Pi <= N)， 表示这N个数字的某一全排列。

输出格式

输出一个整数，表示不同连号区间的数目。

样例输入1

4  
3 2 4 1

样例输出1

7

样例输入2

5  
3 4 2 5 1

样例输出2

9

**参考代码**

**C++**

#include <iostream>

using namespace std;

int a[50001];

int main()

{

int i,j,n,cnt=0;

int x,y;//最大数和最小数

cin>>n;

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

{

cin>>a[i];

}

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

{

x=y=a[i];

for(j=i;j<n;j++)

{

x=max(a[j],x);

y=min(a[j],y);

if(x-y==j-i)

cnt++;

}

}

cout<<cnt;

return 0;

}

**C**

#include<stdio.h>

int main()

{

int s[50005],a,i,min,max,count=0,j;

scanf("%d",&a);

for( i = 0; i < a; i++) {

scanf("%d",&s[i]);

}

for( i = 0; i <a;i++ ){

min=s[i];

max=s[i];

for( j = i; j <a; j++) {

if(min>s[j]){min =s[j];}

if(max<s[j]){max =s[j];}

if((max-min)==(j-i)){

count++;

}

}

}

printf("%d",count);

return 0;

}

**JAVA**

import java.io.\*;

public class Main{

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

//long start=System.currentTimeMillis();

fun();

//System.out.println("\nTime:"+(System.currentTimeMillis()-start)+"ms");

}

public static void fun()throws IOException{

//String filename="test.txt";

//FileInputStream file=new FileInputStream(filename);

//System.setIn(file);

int[] a;

StreamTokenizer st=new StreamTokenizer(new InputStreamReader(System.in));

st.nextToken();

int n=(int)st.nval;

a=new int[n];

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

st.nextToken();

a[i]=(int)st.nval;

}

int min;

int max;

int num=0;

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

min=max=a[l];

for(int r=l;r<n;r++){

if(a[r]<min)min=a[r];

if(a[r]>max)max=a[r];

//

if(max-min==r-l)

num++;

}

}

System.out.print(num);

}

}

PREV-6**历届试题 翻硬币**

时间限制：1.0s   内存限制：256.0MB

关键字贪心

问题描述

小明正在玩一个“翻硬币”的游戏。

桌上放着排成一排的若干硬币。我们用 \* 表示正面，用 o 表示反面（是小写字母，不是零）。

比如，可能情形是：\*\*oo\*\*\*oooo

如果同时翻转左边的两个硬币，则变为：oooo\*\*\*oooo

现在小明的问题是：如果已知了初始状态和要达到的目标状态，每次只能同时翻转相邻的两个硬币,那么对特定的局面，最少要翻动多少次呢？

我们约定：把翻动相邻的两个硬币叫做一步操作，那么要求：

输入格式

两行等长的字符串，分别表示初始状态和要达到的目标状态。每行的长度<1000

输出格式

一个整数，表示最小操作步数。

样例输入1

\*\*\*\*\*\*\*\*\*\*  
o\*\*\*\*o\*\*\*\*

样例输出1

5

样例输入2

\*o\*\*o\*\*\*o\*\*\*  
\*o\*\*\*o\*\*o\*\*\*

样例输出2

1

**参考代码**

**C++**

#include <iostream>

using namespace std;

char reverse(char c)

{

if(c=='\*')

c='o';

else

c='\*';

return c;

}

int main()

{

char a[1000],b[1000];

cin>>a>>b;

int move=0;

for(int i=0;a[i]!='\0';i++)

{

if(a[i]!=b[i])

{

a[i]=reverse(a[i]);

a[i+1]=reverse(a[i+1]);

move++;

}

}

cout<<move<<endl;

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

int turn(char a[], char b[])

{

int i;

int n = 0;

for(i = 0;a[i]!='\0';i++)

{

if(a[i] == b[i])

{

continue;

}

else

{

b[i+1]=(b[i+1]=='\*'?'o':'\*');

n++;

}

}

return n;

}

int main(void)

{

char a[1000];

char b[1000];

gets(a);

gets(b);

printf("%d\n", turn(a, b));

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Main {

public static int cnt = 0;

public static void reverse(char[] origin, int k) {

origin[k] = (origin[k] == '\*') ? 'o' : '\*';

origin[k+1] = (origin[k+1] == '\*') ? 'o' : '\*';

++cnt;

}

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

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

char[] origin = br.readLine().toCharArray();

char[] target = br.readLine().toCharArray();

int k = 0;

while(k < target.length)

if(k+1 < target.length) {

if(origin[k] != target[k] && origin[k+1] == target[k+1]) {

reverse(origin, k);

k = k+1;

} else if(origin[k] != target[k] && origin[k+1] != target[k+1]) {

reverse(origin, k);

k = k+2;

} else if(origin[k] == target[k] && origin[k+1] == target[k+1])

k = k+2;

else

k = k+1;

} else

k = k+1;

System.out.println(cnt);

}

}

PREV-5**历届试题 错误票据**

时间限制：1.0s   内存限制：256.0MB

关键字 排序

问题描述

某涉密单位下发了某种票据，并要在年终全部收回。

每张票据有唯一的ID号。全年所有票据的ID号是连续的，但ID的开始数码是随机选定的。

因为工作人员疏忽，在录入ID号的时候发生了一处错误，造成了某个ID断号，另外一个ID重号。

你的任务是通过编程，找出断号的ID和重号的ID。

假设断号不可能发生在最大和最小号。

输入格式

要求程序首先输入一个整数N(N<100)表示后面数据行数。

接着读入N行数据。

每行数据长度不等，是用空格分开的若干个（不大于100个）正整数（不大于100000），请注意行内和行末可能有多余的空格，你的程序需要能处理这些空格。

每个整数代表一个ID号。

输出格式

要求程序输出1行，含两个整数m n，用空格分隔。

其中，m表示断号ID，n表示重号ID

样例输入1

2  
5 6 8 11 9   
10 12 9

样例输出1

7 9

样例输入2

6  
164 178 108 109 180 155 141 159 104 182 179 118 137 184 115 124 125 129 168 196  
172 189 127 107 112 192 103 131 133 169 158   
128 102 110 148 139 157 140 195 197  
185 152 135 106 123 173 122 136 174 191 145 116 151 143 175 120 161 134 162 190  
149 138 142 146 199 126 165 156 153 193 144 166 170 121 171 132 101 194 187 188  
113 130 176 154 177 120 117 150 114 183 186 181 100 163 160 167 147 198 111 119

样例输出2

105 120

**参考代码**

**C++**

#include<cstdio>

#include<cstring>

#include<cmath>

#include<cstdlib>

#include<algorithm>

#include<iostream>

#include<string>

using namespace std;

int a[100005];

int main(){

int n;

while(cin>>n){

char c[1005];

int b[105],t=0;

memset(a,0,sizeof(a));

getchar();

while(n--){

gets(c);

int tc=strlen(c);

c[tc]=' ',c[tc+1]=0;

int tem=0;

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

if(c[i]!=' '){

tem=tem\*10+c[i]-'0';

}

else{

a[tem]++;

b[t++]=tem;

tem=0;

}

}

}

int dh,ch;

sort(b,b+t);

for(int i=b[0];i<b[t-1];i++)

if(a[i]==0)

dh=i;

else if(a[i]==2)

ch=i;

cout<<dh<<' '<<ch<<endl;

}

return 0;

}

**C**

#include <stdio.h>

int main()

{

int a[10001]={0};

long m,min=100000,max=0,i,n;

char c;

scanf("%d",&n);

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

while(1)

{

scanf("%ld",&m);

if(m>max) max=m;

if(m<min) min=m;

a[m]++;

c=getchar();

if(c!=' ') break;

}

for(i=min;i<=max;i++)

{

if(a[i]==0) printf("%ld ",i);

if(a[i]==2) m=i;

}

printf("%ld",m);

return 0;

}

**JAVA**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

public class Main {

public static void main(String[] args) throws Exception {

int duan = 0;

int chong = 0;

List<Integer> list = new ArrayList();

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

String line = br.readLine();

int len = Integer.parseInt(line);

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

{

String l = br.readLine();

String[] s = l.split(" ");

for(int j = 0; j < s.length; j++)

{

list.add(Integer.parseInt(s[j]));

}

}

for(int i = 0 ; i < list.size(); i++)

{

for(int j = 0; j < list.size(); j++)

{

int a = list.get(i);

int b = list.get(j);

int temp = 0;

if(a < b)

{

temp = a;

// a = b;

// b = temp;

list.set(i, b);

list.set(j, temp);

}

}

}

for(int i = 0 ; i < list.size() - 1; i++)

{

int a = list.get(i);

int b = list.get(i + 1);

if(a == b)

chong = a;

if(b - a > 1)

duan = b -1;

}

System.out.println(duan + " " + chong);

}

}

PREV-4 **历届试题 剪格子**

时间限制：1.0s   内存限制：256.0MB

关键字 搜索

问题描述

如下图所示，3 x 3 的格子中填写了一些整数。

+--\*--+--+  
|10\* 1|52|  
+--\*\*\*\*--+  
|20|30\* 1|  
\*\*\*\*\*\*\*--+  
| 1| 2| 3|  
+--+--+--+

我们沿着图中的星号线剪开，得到两个部分，每个部分的数字和都是60。

本题的要求就是请你编程判定：对给定的m x n 的格子中的整数，是否可以分割为两个部分，使得这两个区域的数字和相等。

如果存在多种解答，请输出包含左上角格子的那个区域包含的格子的最小数目。

如果无法分割，则输出 0。

输入格式

程序先读入两个整数 m n 用空格分割 (m,n<10)。

表示表格的宽度和高度。

接下来是n行，每行m个正整数，用空格分开。每个整数不大于10000。

输出格式

输出一个整数，表示在所有解中，包含左上角的分割区可能包含的最小的格子数目。

样例输入1

3 3  
10 1 52  
20 30 1  
1 2 3

样例输出1

3

样例输入2

4 3  
1 1 1 1  
1 30 80 2  
1 1 1 100

样例输出2

10

**参考代码**

**C++**

#include<cstdio>

#include<cstring>

#include<cmath>

#include<cstdlib>

#include<algorithm>

#include<iostream>

#include<string>

#include<queue>

using namespace std;

int a[15][15];

int m,n;

int mx,sum;

int mi[][2]={{0,1},{0,-1},{1,0},{-1,0}};

struct ki{

int x,y;

int sum;

int tep;

};

void dfs(int x,int y,int s,int tep){

if(s>sum/(2.0))

return ;

else if(s==sum/(2.0)){

if(tep<mx)

mx=tep;

// printf("s=%d\n",s);

}

else{

for(int i=0;i<4;i++){

int tx=x+mi[i][0];

int ty=y+mi[i][1];

if(tx<0||ty<0||tx>=n||ty>=m)

continue;

if(a[tx][ty]!=-1){

int tem=a[tx][ty];

a[tx][ty]=-1;

// printf("tx=%d ty=%d s=%d tep=%d\n",tx,ty,s,tep);

dfs(tx,ty,s+tem,tep+1);

a[tx][ty]=tem;

}

}

}

}

int main(){

while(cin>>m>>n){

sum=0;

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

for(int k=0;k<m;k++){

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

sum+=a[i][k];

}

mx=10000;

int tem=a[0][0];

a[0][0]=-1;

dfs(0,0,tem,1);

printf("%d\n",mx==10000?0:mx);

}

return 0;

}

**C**

#include <stdio.h>

#define N 10

int num[N][N];

int tag[N][N] = {0};

int m, n;

int r = 100;

int find(int i, int j, int t, int ntag[][N])

{

int count = 0;

if (i < 0 || i >= n || j < 0 || j >= m || ntag[i][j] == 1)

return 0;

ntag[i][j] = 1;

if (tag[i][j] != t)

return 0;

count++;

count += find(i - 1, j, t, ntag);

count += find(i + 1, j, t, ntag);

count += find(i, j - 1, t, ntag);

count += find(i, j + 1, t, ntag);

return count;

}

int isbad()

{

int i, j, k = 0,ge2;

int t = tag[0][0];

int ntag1[N][N] = {0};

int ntag2[N][N] = {0};

int ge1 = find(0, 0, t, ntag1);

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

{

for (j = 0; j < m; j++)

{

if (tag[i][j] != t)

{

k = 1;

break;

}

}

if (k == 1)

break;

}

if (i == n && j == m)

return 0;

ge2 = find(i, j, tag[i][j], ntag2);

return ge1 + ge2 != m \* n;

}

int bad(int i, int j)

{

int b;

if (i < 0 || i >= n || j < 0 || j >= m || tag[i][j] == 1)

return 1;

tag[i][j] = 1;

b = isbad();

tag[i][j] = 0;

return b;

}

void go(int i, int j, int k, int count)

{

if (bad(i, j) || count < num[i][j])

return;

k++;

if (count == num[i][j])

{

if (r > k)

r = k;

return;

}

tag[i][j] = 1;

count -= num[i][j];

go(i - 1, j, k, count);

go(i + 1, j, k, count);

go(i, j - 1, k, count);

go(i, j + 1, k, count);

tag[i][j] = 0;

}

int main()

{

int i, j;

int half = 0;

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

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

for (j = 0; j < m; j++)

{

scanf("%d", &num[i][j]);

half += num[i][j];

}

if (half % 2 == 0 && half >= num[0][0] \* 2)

{

half /= 2;

go(0, 0, 0, half);

}

if (r == 100)

r = 0;

printf("%d", r);

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

static int sum,half,m,n;

static int[][] map;

static boolean[][] flag;

static int count=100;

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

n=sc.nextInt();

m=sc.nextInt();

map=new int[m][n];

flag=new boolean[m][n];

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

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

map[i][j]=sc.nextInt();

sum+=map[i][j];

}

if(sum%2!=0)

System.out.println(0);

else{

half=sum/2;

dfs(0,0,1,0);

System.out.println(count==100?0:count);

}

}

private static void dfs(int i, int j, int step,int s) {

if(i<0 || i>=m || j<0 || j>=n ){

return;

}

if(flag[i][j]==true)

return;

if(s>half){

flag[i][j]=false;

return;

}

else{

s+=map[i][j];

flag[i][j]=true;

if(s==half){

if(step<count)

count=step;

}

else{

dfs(i+1,j,step+1,s);

dfs(i,j+1,step+1,s);

dfs(i-1,j,step+1,s);

dfs(i,j-1,step+1,s);

flag[i][j]=false;

}

}

}

}

PREV-3 **历届试题 带分数**

时间限制：1.0s   内存限制：256.0MB

关键字 搜索

问题描述

100 可以表示为带分数的形式：100 = 3 + 69258 / 714。

还可以表示为：100 = 82 + 3546 / 197。

注意特征：带分数中，数字1~9分别出现且只出现一次（不包含0）。

类似这样的带分数，100 有 11 种表示法。

输入格式

从标准输入读入一个正整数N (N<1000\*1000)

输出格式

程序输出该数字用数码1~9不重复不遗漏地组成带分数表示的全部种数。

注意：不要求输出每个表示，只统计有多少表示法！

样例输入1

100

样例输出1

11

样例输入2

105

样例输出2

6

**参考代码**

**C++**

#include<iostream>

#include<cstring>

using namespace std;

int N,ans,digit,flag,full[9];

void div(int m)

{

while(m)

{

if(m%10!=0)

full[m%10-1]=1;

m/=10;

digit++;

}

}

bool check(int \*f)

{

int j;

for(j=0;j<9;j++)

if(!f[j])

return false;

return true;

}

void DFS(int a,int b,int c)

{

digit=0;

flag=0;

div(a);

div(b);

div(c);

if(digit>9){memset(full,0,sizeof(full));return ;}

if(check(full)){ans++;memset(full,0,sizeof(full));}

else memset(full,0,sizeof(full));

DFS(a,(c+1)\*b/c,c+1);

return ;

}

int main()

{

int i;

while(cin>>N)

{

ans=0;

memset(full,0,sizeof(full));

for(i=2;i<N-1;i++)

DFS(i,N-i,1);

cout<<ans<<endl;

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

typedef struct Interval

{

int pre;

int rear;

int satisfy;

}Interval;

Interval interval[7][5];

int count=0;

//初始化

void Init()

{

int i,j;

int value;

for(i=1;i<7;i++)

{

value=i;

for(j=1;j<5;j++)

{

interval[i][j].pre=value++;

interval[i][j].rear=value;

}

}

}

//数组初始化为0

void InitZero(int \*sign)

{

int i;

sign[0]=1;

for(i=1;i<10;i++)

{

sign[i]=0;

}

}

//将一个数的各个位上拆分，并在相应的位上赋值1

int Split(int \*sign,int value)

{

int index;

while(value)

{

index=value%10;

if(sign[index]==0) sign[index]=1;

else return 1;

value/=10;

}

return 0;

}

//计算一个数的位数

int CountBit(int value)

{

int n=0;

while(value)

{

n++;

value/=10;

}

return n;

}

//将一个整型数组转换成一个整数

int CreateInteger(int \*data,int n)

{

int i;

int value=0;

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

{

value=value\*10+data[i];

}

return value;

}

//检查是否每个数都用到

int Check(int \*sign)

{

int i;

for(i=1;i<10;i++)

{

if(sign[i]==0) return 0;

}

return 1;

}

//复制

void Copy(int \*sign,int \*temp\_sign)

{

int i;

for(i=0;i<10;i++)

{

temp\_sign[i]=sign[i];

}

}

//创建一个n位数的整数

void CreateNBitNumber(int \*sign,int \*data,int n,int m,int value,int value3)

{

if(n==m)

{

int value1=CreateInteger(data,n);

int value2=value1\*value;

int temp\_sign[10];

Copy(sign,temp\_sign);

if(!Split(temp\_sign,value2) && Check(temp\_sign))

{

count++;

}

}

else

{

int i;

for(i=1;i<10;i++)

{

if(sign[i]==0)

{

sign[i]=1;

data[m]=i;

CreateNBitNumber(sign,data,n,m+1,value,value3);

sign[i]=0;

}

}

}

}

//求出解

void Create(int value)

{

int i,j;

int sign[10];

int result;

int result\_n;

int n;

for(i=3;i<value;i++)

{

InitZero(sign);

if(Split(sign,i)) continue;

result=value-i;

result\_n=CountBit(result);

n=CountBit(i);

for(j=1;j<5;j++)

{

if( ((interval[result\_n][j].pre+j)==(9-n)) || ((interval[result\_n][j].rear+j)==(9-n)))

{

int data[5];

CreateNBitNumber(sign,data,j,0,result,i);

}

}

}

}

int main()

{

int value;

scanf("%d",&value);

Init();

Create(value);

printf("%d\n",count);

return 0;

}

**JAVA**

import java.util.Scanner;

/\*\*

\* http://lx.lanqiao.org/problem.page?gpid=T26

\* Created by revintec on 14-5-5.

\*/

public class Main{

static int getBits(int k){

int bits=0;

while(k>0){

int a=k%10;k/=10;

if(a==0) return -1;

int b=1<<a;

if((bits&b)!=0) return -1;

bits|=b;

}return bits;

}

static int getBits2(int b){

int lo=9;

for(int i=1;i<1<<10;i<<=1)

if((b&i)!=0) --lo;

return lo;

}

public static void main(String[] args){

// System.clearProperty("m.TEST");

Scanner sc=System.getProperty("m.TEST")==null?new Scanner(System.in):

new Scanner("999988");

int sx=sc.nextInt();

int[] ts={0,

10,

100,

1000,

10000,

100000,

1000000,

10000000,

100000000,

1000000000,

};

int so=0;

for(int a=1;a<sx;++a){

int bits=getBits(a);

if(bits==-1) continue;

for(int b=1;;++b){

int bts=getBits(b);

if((bits&bts)!=0) continue;

bts|=bits;

long c=b\*(sx-a);

if(c>=ts[getBits2(bts)]) break;

if((getBits((int)c)^bts)==1022)

++so;

}

}System.out.println(so);

}

}

PREV-2**历届试题 打印十字图**

时间限制：1.0s   内存限制：256.0MB

关键字 文字图形

问题描述

小明为某机构设计了一个十字型的徽标（并非红十字会啊），如下所示：

..$$$$$$$$$$$$$..  
..$...........$..  
$$$.$$$$$$$$$.$$$  
$...$.......$...$  
$.$$$.$$$$$.$$$.$  
$.$...$...$...$.$  
$.$.$$$.$.$$$.$.$  
$.$.$...$...$.$.$  
$.$.$.$$$$$.$.$.$  
$.$.$...$...$.$.$  
$.$.$$$.$.$$$.$.$  
$.$...$...$...$.$  
$.$$$.$$$$$.$$$.$  
$...$.......$...$  
$$$.$$$$$$$$$.$$$  
..$...........$..  
..$$$$$$$$$$$$$..

对方同时也需要在电脑dos窗口中以字符的形式输出该标志，并能任意控制层数。

输入格式

一个正整数 n (n<30) 表示要求打印图形的层数。

输出格式

对应包围层数的该标志。

样例输入1

1

样例输出1

..$$$$$..  
..$...$..  
$$$.$.$$$  
$...$...$  
$.$$$$$.$  
$...$...$  
$$$.$.$$$  
..$...$..  
..$$$$$..

样例输入2

3

样例输出2

..$$$$$$$$$$$$$..  
..$...........$..  
$$$.$$$$$$$$$.$$$  
$...$.......$...$  
$.$$$.$$$$$.$$$.$  
$.$...$...$...$.$  
$.$.$$$.$.$$$.$.$  
$.$.$...$...$.$.$  
$.$.$.$$$$$.$.$.$  
$.$.$...$...$.$.$  
$.$.$$$.$.$$$.$.$  
$.$...$...$...$.$  
$.$$$.$$$$$.$$$.$  
$...$.......$...$  
$$$.$$$$$$$$$.$$$  
..$...........$..  
..$$$$$$$$$$$$$..

提示

请仔细观察样例，尤其要注意句点的数量和输出位置。

**参考代码**

**C++**

#include <stdio.h>

#include <malloc.h>

#include <string.h>

int main()

{

int n,w,h,l,i,j,x,y,m;

char \*arry;

scanf("%d",&n);

w=h=5+n\*4;

arry=(char \*)malloc(w\*h);

memset(arry,'.',w\*h);

/\*

for(i=n\*2;i<w-n\*2;i++)//画基础图形

{

arry[i+((n+1)\*2)\*w]='$';

arry[i\*w+(n+1)\*2]='$';

}\*/

for(m=0;m<=n;m++)//画没一层

{

for(i=(m+1)\*2;i<w-(m+1)\*2;i++)//四边

{

x=m\*2;

y=i;

arry[x+y\*w]='$';

x=w-m\*2-1;

y=i;

arry[x+y\*w]='$';

x=i;

y=m\*2;

arry[x+y\*w]='$';

x=i;

y=w-m\*2-1;

arry[x+y\*w]='$';

}

for(i=m\*2;i<=(m+1)\*2;i++)//角

{

x=i;

y=(m+1)\*2;

arry[x+y\*w]='$';

x=(m+1)\*2;

y=i;

arry[x+y\*w]='$';

x=w-i-1;

y=(m+1)\*2;

arry[x+y\*w]='$';

x=w-(m+1)\*2-1;

y=i;

arry[x+y\*w]='$';

x=i;

y=h-(m+1)\*2-1;

arry[x+y\*w]='$';

x=(m+1)\*2;

y=h-i-1;

arry[x+y\*w]='$';

x=w-i-1;

y=h-(m+1)\*2-1;

arry[x+y\*w]='$';

x=w-(m+1)\*2-1;

y=h-i-1;

arry[x+y\*w]='$';

}

}

for(i=0;i<h;i++)

{

for(j=0;j<w;j++)

{

printf("%c",arry[i\*w+j]);

}

printf("\n");

}

return 0;

}

**C**

#include <stdio.h>

#include <malloc.h>

#include <string.h>

int main()

{

int n,w,h,l,i,j,x,y,m;

char \*arry;

scanf("%d",&n);

w=h=5+n\*4;

arry=(char \*)malloc(w\*h);

memset(arry,'.',w\*h);

/\*

for(i=n\*2;i<w-n\*2;i++)//画基础图形

{

arry[i+((n+1)\*2)\*w]='$';

arry[i\*w+(n+1)\*2]='$';

}\*/

for(m=0;m<=n;m++)//画没一层

{

for(i=(m+1)\*2;i<w-(m+1)\*2;i++)//四边

{

x=m\*2;

y=i;

arry[x+y\*w]='$';

x=w-m\*2-1;

y=i;

arry[x+y\*w]='$';

x=i;

y=m\*2;

arry[x+y\*w]='$';

x=i;

y=w-m\*2-1;

arry[x+y\*w]='$';

}

for(i=m\*2;i<=(m+1)\*2;i++)//角

{

x=i;

y=(m+1)\*2;

arry[x+y\*w]='$';

x=(m+1)\*2;

y=i;

arry[x+y\*w]='$';

x=w-i-1;

y=(m+1)\*2;

arry[x+y\*w]='$';

x=w-(m+1)\*2-1;

y=i;

arry[x+y\*w]='$';

x=i;

y=h-(m+1)\*2-1;

arry[x+y\*w]='$';

x=(m+1)\*2;

y=h-i-1;

arry[x+y\*w]='$';

x=w-i-1;

y=h-(m+1)\*2-1;

arry[x+y\*w]='$';

x=w-(m+1)\*2-1;

y=h-i-1;

arry[x+y\*w]='$';

}

}

for(i=0;i<h;i++)

{

for(j=0;j<w;j++)

{

printf("%c",arry[i\*w+j]);

}

printf("\n");

}

return 0;

}

**JAVA**

import java.util.Scanner;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

fd(sc.nextInt());

}

public static void fd(int n){

int len=5+4\*n;

String a[][]=new String[len][len];

int b=len/2;

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

for(int j=0;j<len;j++){

a[i][j]=".";

}}

for(int i=b-2;i<b+2;i++){

a[i][b]="$";

a[b][i]="$";

}

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

for(int i=b-2-2\*m;i<(b+3+2\*m);i++){

a[b - (4+m\*2)][i] = "$";

a[b + (4+m\*2)][i] = "$";

a[i][b - (4+m\*2)] = "$";

a[i][b + (4+m\*2)] = "$";

}

}

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

for (int i = b - (3+m\*2); i <= b + 3+m\*2; i++) {

a[b - (2+m\*2)][i] = "$";

a[b + 2+m\*2][i] = "$";

a[i][b + 2+m\*2] = "$";

a[i][b - (2+m\*2)]= "$";

}

a[b - (2+m\*2)][b - (1+m\*2)] = ".";

a[b - (2+m\*2)][b + 1+m\*2] = ".";

a[b + 2+m\*2][b - (1+m\*2)] = ".";

a[b + 2+m\*2][b + 1+m\*2] = ".";

a[b - (1+m\*2)][b - (2+m\*2)] = ".";

a[b + 1+m\*2][b - (2+m\*2)] = ".";

a[b - (1+m\*2)][b + 2+m\*2] = ".";

a[b + 1+m\*2][b + 2+m\*2] = ".";

}

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

for (int j = 0; j < len; j++) {

System.out.print(a[i][j]);

}

System.out.println();

}

}

}

PREV-1**历届试题 核桃的数量**

时间限制：1.0s   内存限制：256.0MB

关键字 最小公倍数

问题描述

小张是软件项目经理，他带领3个开发组。工期紧，今天都在加班呢。为鼓舞士气，小张打算给每个组发一袋核桃（据传言能补脑）。他的要求是：

1. 各组的核桃数量必须相同

2. 各组内必须能平分核桃（当然是不能打碎的）

3. 尽量提供满足1,2条件的最小数量（节约闹革命嘛）

输入格式

输入包含三个正整数a, b, c，表示每个组正在加班的人数，用空格分开（a,b,c<30）

输出格式

输出一个正整数，表示每袋核桃的数量。

样例输入1

2 4 5

样例输出1

20

样例输入2

3 1 1

样例输出2

3

**参考代码**

**C++**

#include <stdio.h>

int main(int argc, char \*argv[]) {

int a,b,c,i;

int all;

scanf("%d %d %d",&a,&b,&c);

all=a\*b\*c;

for(i=1;i<=all;i++)

{

if(i%a==0&&i%b==0&&i%c==0)

{

printf("%d",i);

break;

}

}

return 0;

}

**C**

#include <stdio.h>

#include <stdlib.h>

//最小公倍数

int LCM(int num1,int num2,int num3)

{

int value=num1;

while(value%num1!=0||value%num2!=0||value%num3!=0)

{

value+=num1;

}

return value;

}

int main()

{

int num1,num2,num3;

scanf("%d%d%d",&num1,&num2,&num3);

printf("%d\n",LCM(num1,num2,num3));

return 0;

}

**JAVA**

import java.io.\*;

public class Main {

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

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

String str[] = br.readLine().split(" ");

int[] arr = new int[3];

for (int a = 0; a < arr.length; a++) {

arr[a] = Integer.parseInt(str[a]);

}

for (int a = 1; a < 27000; a++) {

if (a % arr[0] == 0 && a % arr[1] == 0 && a % arr[2] == 0) {

System.out.print(a);

break;

}

}

}

}