

C4-2021级航类-第四次上机

这是 2021级-航类-C语言程序设计 的第四次上机。

- 上机时间2022/03/30 19:00 - 2022/03/30 20:30，一个半小时。
- 总共10道题，编号为A~J，部分题目由课件例题改编而来。
- 题目不一定按难度排序，同学们可以按照通过人数答题；或者提前读完所有题目，按照自己擅长题目的情况答题。
- 各题分值分布为如下，总分102分，满分100分（超过100分按100分计算）：

序号	A	B	C	D	E	F	G	H	I	J
分值	20	20	20	10	10	10	3	3	3	3

- 请严肃比赛，严禁抄袭他人代码，课程组会在比赛结束后进行代码查重并给予警告。

总体规划与课件题目提供：方宁老师、李可老师

组题：Arthas、zym、助教头子cbd、爱吃猪脚的猪脚

更多题目作者：2020助教dcy、宋友老师、wwh、ljh、czy、呱呱泡蛙、dwy

A <math.h>

题目背景

当我们需要使用一些常见的数学公式时，可以利用 C 语言常用的函数库 `math.h`，在程序的开头部分加上一行 `#include<math.h>` 就可以使用其中的函数了。下面是一些 `math.h` 中常见的函数：

（参考自[常用数学函数](#)）

1. 基本运算

```
double fabs(double); //求实型的绝对值
double fmod(double, double); //返回两参数相除的余数
double remainder(double, double); //计算浮点除法运算的带符号余数
double fmax(double, double); //确定两个浮点值的较大者
double fmin(double, double); //确定两个浮点值的较小者
double fdim(double, double); //确定两个浮点值的非负数差
```

2. 指数与对数函数

```
double exp(double); //求取自然数e的幂
double exp2(double); //求取2的幂
double log(double); //以e为底的对数
double log10(double); //以10为底的对数
double log2(double); //以2为底的对数
```

3. 幂函数

```
double pow(double x, double y); // 计算x的y次幂
double sqrt(double); // 开平方根
double cbrt(double); // 开立方根
double hypot(double x, double y); // 已知直角三角形两个直角边长度，求斜边长度
```

4. 三角与反三角函数

```
double sin(double); // 正弦
double cos(double); // 余弦
double tan(double); // 正切
double asin(double); // 结果介于 $[-\pi/2, \pi/2]$ 
double acos(double); // 结果介于 $[0, \pi]$ 
double atan(double); // 反正切（主值），结果介于 $[-\pi/2, \pi/2]$ 
double atan2(double, double); // 反正切（整圆值），结果介于 $[-\pi, \pi]$ 
```

5. 双曲与反双曲函数

```
double sinh(double);
double cosh(double);
double tanh(double);
double asinh(double);
double acosh(double);
double atanh(double);
```

6. 误差及gamma函数

```
double erf(double); // 计算误差函数
double erfc(double); // 计算补误差函数
double tgamma(double); // 计算伽马函数
double lgamma(double); // 计算伽马函数的自然对数
```

7. 临近整数的浮点运算

```
double ceil(double); // 取上整，返回不比x小的最小整数
double floor(double); // 取下整，返回不比x大的最大整数，即高斯函数 $[x]$ 
double trunc(double); // 取整到绝对值不大于给定值的最接近整数
double round(double); // 取整到最接近的整数，在相邻整数正中间时取远离零的数
```

8. 浮点数操作函数

```
double frexp(double value,int *exponent);//标准化浮点数, value=x*2^exponent, 已知
value 求 x,exponent
//把浮点数分解成尾数和指数。返回值是尾数, 并将指数存入 exponent 中。
//其中尾数 x 取值在 (-1, -0.5] 或 [0.5, 1) 或 0。
double ldexp(double x,int exponent);//计算x (2的指数幂), 已知 x,exponent 求f
//这个函数刚好跟上面那个frexp函数功能相反, 它的返回值是x倍的2的指数幂
double modf(double value,double *iptr);//拆分value值, 返回它的小数部分, iptr指向整数部分。
```

题目描述

计算

$$S = \frac{(\arccos(\sin x))^{\ln(1+|\sinh y|)}}{2 + \cos z}$$

(题目保证输入输出以及中间结果都在 `double` 范围内)

输入格式

输入一行三个数字 x, y, z 。

输出格式

输出一行一个数字 S , 保留两位小数。

样例输入1

2 3.5 4

样例输出1

0.07

样例输入2

1 2 3

样例输出2

0.42

copyright: Chenyang Du

题目搬运: Bodan Chen

题面整理: 呱呱泡蛙

B 宋老师的名次预测2.0

题目描述

还记得C1 甲贺忍蛙的名次预测 吗？这次来了个升级版2.0！选手人次增加到了30人，且实际结果是这30人依次获得了第1到第30名，请你判断预测对了多少人吧

输入格式

输入共 $n + 1$ 行；
第一行一个正整数 n ，表示接下来有 n 组预测；
接下来 n 行每行为30个以空格分隔的整数，依次是对30位选手预测的名次；

输出格式

对于每组预测输出一行，记猜对人数为num

- 如果num大于20，输出 *****
- 如果num不大于20但大于10，输出 ***
- 如果num不大于10但大于5，输出 **
- 否则，输出 #

输入样例

```
3
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
30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
15 2 7 6 3 23 14 30 9 8 11 29 10 1 4 19 20 16 12 27 13 22 21 24 17 26 18 25 28 5
```

输出样例

```
*****
#
**
```

样例解释

对于输入的三组预测，猜对的人数分别是30、0、6，所以输出如上

数据范围

$1 \leq n \leq 100$
AUTHOR:Songyou

C 滴滴滴递归

题目介绍

给出一个简单的递归函数如下

$$fun(a, b) = \begin{cases} 1 & \text{if } b = 0; \\ fun(a * a, b / 2) & \text{else if } b \% 2 = 0; \\ fun(a * a, b / 2) * a & \text{else.} \end{cases}$$

现给出 a, b ，请你计算 $fun(a, b)$ 的值；

输入格式

多组数据输入，输入组数不超过 200；

每组数据一行，两个用空格分隔的整数 a, b ， $a > 0, b \geq 0$ ，保证输出结果在 `int` 范围内。

输出格式

对于每组数据，输出一行，表示 $fun(a, b)$ 的值。

输入样例

```
1 2
2 1
2 2
```

输出样例

```
1
2
4
```

HINT

参考课件P5 例5-12 阿克曼函数的递归写法，本题的递归函数可以在如下代码中补充

```
int fun(int a, int b)
{
    if (b == 0)
    {
        // do something
    }
}
```

```
else if (b % 2 == 0)
{
    // do something
}
else
{
    // do something
}
}
```

ps

大家可以思考一下这个递归函数实际上计算的是什么，可以根据输出结果来猜测然后验证。

AUTHOR: Bodan Chen

D 求差向量的无穷范数

题目背景

自定义一个函数，计算两个向量的差向量的无穷范数。

向量 X 的无穷范数定义为

$$||X||_{\infty} = \max_{1 \leq i \leq n} |x_i|$$

请大家体会 “利用函数实现结构化编程，可以提高程序开发效率”。

题目描述

输入5个三维空间向量，按照要求的顺序输出它们两两之间差向量的无穷范数。

输入格式

共5行，每行3个浮点数 x, y, z ($-1000 < x, y, z < 1000$)

对于第 i 行， x_i, y_i, z_i 代表第 i 个点的三维空间向量，数字之间用空格隔开。

输出格式

共8行，从第1行到第8行分别输出：向量5与向量4，向量3与向量1，向量1与向量5，向量4与向量2，向量3与向量5，向量2与向量1，向量3与向量4，向量1与向量4的差向量的无穷范数。结果保留3位小数。

输入样例1

```
0 0 0
1 2 3
5 1 3
6 2 4
8 1 0
```

输出样例1

```
4.000
5.000
8.000
5.000
3.000
3.000
1.000
6.000
```

样例解释

向量5与向量4的差向量为 $(2, -1, -4)$,其无穷范数为向量中绝对值最大元素的绝对值，输出为4.000

向量3与向量1的差向量为 $(5, 1, 3)$,其无穷范数为向量中绝对值最大元素的绝对值，输出为5.000

向量1与向量5的差向量为 $(-8, -1, 0)$,其无穷范数为向量中绝对值最大元素的绝对值，输出为8.000

以此类推。

输入样例2

```
1.23 256 235.2
2.5 895.2 3.7
999.9 999.9 91.8
63.235 56.235 25.142
21 76 15
```

输出样例2

```
42.235
998.670
220.200
838.965
978.900
639.200
943.665
210.058
```

HINT

如果你不知道怎么定义函数，可以参考下面的 $a + b$ 例程，相信能够帮助到你

```
#include<stdio.h>
int solve(int x,int y)//此处int为返回值的类型，x和y分别对应a和b，代表传递来的参数。
{
    int z=x+y;
    return z;//用return返回计算结果
}
int main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    int c=solve(a,b);
    printf("%d",c);
    return 0;
}

// 对于本题，可以用三个数组x[],y[],z[]分别存储坐标点的坐标
// 例如第一个点的坐标就是(x[0],y[0],z[0])，当然也可以是(x[1],y[1],z[1])，随你所好
// 定义一个函数printNorm()，函数接受两个int参数i, j
// 返回值为第i个向量和第j个向量的差向量的无穷范数

double x[10], y[10], z[10];

double printNorm(int i, int j)
{
    // code here
}
```

AUTHOR: wwh

E 多项式相加2022

题目描述

一元多项式的定义如下：

- 设 c_0, c_1, \dots, c_n 都是数域 F 中的数， n 是非负整数，那么表达式

$$c_n \times x^n + c_{n-1} \times x^{n-1} + \dots + c_2 \times x^2 + c_1 \times x + c_0$$

就是数域 F 上关于变量 x 的多项式或一元多项式。

- 其中， $c_k \times x^k$ ($1 \leq k \leq n$) 代表该一元多项式中的一个项， c_k 是该项的系数， k 是该项的指数。

现在给定两个整数数域上关于变量 x 的一元多项式 $f(x)$ 和 $g(x)$ ，请你求出二者相加后产生的一元多项式 $f(x) + g(x)$ ，并要求不再输出系数为 0 的项。

输入格式

第一行两个整数 N, M ($1 \leq N, M \leq 100000$)，分别代表 $f(x)$ 和 $g(x)$ 的项数。

第二行 $2 \times N$ 个整数，第 $2 \times i - 1$ 和 $2 \times i$ 个整数分别代表 $f(x)$ 中第 i 项的系数 a_i 和指数 s_i ， a_i 和 s_i 在 `int` 范围内，且 $a_i \neq 0$ 。

第三行 $2 \times M$ 个整数，第 $2 \times j - 1$ 和 $2 \times j$ 个整数分别代表 $g(x)$ 中第 j 项的系数 b_j 和指数 t_j ， b_j 和 t_j 在 `int` 范围内，且 $b_j \neq 0$ 。

保证多项式 $f(x)$ 和 $g(x)$ 中的项按指数**严格降序**给定。

输出格式

输出一行，包含偶数个整数，第 $2 \times k - 1$ 和 $2 \times k$ 个整数分别代表 $f(x) + g(x)$ 中第 k 项的系数和指数，并以指数**严格降序**输出。

样例输入

```
3 2
6 3 3 2 9 1
-6 3 -5 1
```

样例输出

```
3 2 4 1
```

样例解释

根据题意可知

$$f(x) = 6 \times x^3 + 3 \times x^2 + 9 \times x$$

$$g(x) = -6 \times x^3 - 5 \times x$$

因而

$$f(x) + g(x) = 3 \times x^2 + 4 \times x$$

即输出 `3 2 4 1`。

F 简单的基物数据处理

题目描述

基础物理实验是北航的一门特色工科课程，小林今天完成了代号1091的实验，并得到了一系列数据，但他今天很忙，所以想拜托你帮忙处理一下实验数据。

实验数据由一系列浮点数组成，小林想请你帮忙算一下这些数的平均值 avg 和A类不确定度 u_a 。

假设总共有 n 个浮点数 x_i ，则 avg 和 u_a 的计算公式如下：

$$avg = \frac{1}{n} \sum_{i=1}^n x_i$$
$$u_a = \sqrt{\frac{1}{n \times (n - 1)} \sum_{i=1}^n (x_i - avg)^2}$$

输入

一行，由多个浮点数组成，以空格分隔，文件结束符 `EOF` 为结尾。

输出

一行，由两个浮点数组成，以一个空格分隔，分别为这些数的平均值 avg 和A类不确定度 u_a ，结果保留 4 位小数。

输入样例

```
1.0 2.0 3.0 4.0 5.0
```

输出样例

```
3.0000 0.7071
```

数据范围

$0 < x_i \leq 1000$ ，输入共有 n 个浮点数，且 n 满足 $1 < n \leq 1000$ 。

HINT

A类不确定度 u_a 的定义：用对观测列进行统计分析的方法来评定标准不确定度，称为不确定度A类评定，所得到的相应标准不确定度称为A类不确定度分量。

可以采取如下方式输入数据：

```
while(scanf("%lf",&x[i])!=EOF) i++;
```

这道题的代码建议好好保存，下学期的基物实验中可能会有大用-。

AUTHOR: ljh

G 来个神奇日历

题目描述

zym 希望拥有一个神奇的日历，能够查到某年的第 d 天是当年的几月几日。

你可以帮助她实现这个愿望吗？

输入格式

共 $n + 1$ 行。

第 1 行输入一个正整数 n ，表示待查询的日期有 n 组。

接下来 n 行每行输入两个整数 y_i 和 d_i ，其中 y_i 表示年份， d_i 表示要查询的是当年的第 d_i 天。

输出格式

共 n 行。

每行输出两个整数 $month_i$ 和 day_i ，中间用一个空格分开，

表示第 i 个查询结果是当年的 $month_i$ 月 day_i 日。

输入样例

```
1
2022 82
```

输出样例

```
3 23
```

样例解释

2022 年的第 82 天是这年的 3 月 23 日

数据范围

$$1 \leq n \leq 100$$

$$1900 \leq y_i \leq 2025$$

$$1 \leq d_i \leq 366$$

输入数据保证每一组 y_i 和 d_i 都是合法的

HINT

1. 满足下面条件之一的被称为闰年，闰年的 2 月有 29 天。

- 年份是 4 的整数倍，而且不是 100 的整数倍。
 - 年份是 400 的整数倍。
2. 把每个月的天数存到一个数组也许会方便很多。
 3. 每年的第一天是 1 月 1 日。

AUTHOR: zym

H 全排列组合

问题描述

输入整数 m 和 n ，按字典序生成从 m 到 n 之间的所有整数的全排列。

输入格式

单组数据读入。一组数据包含两个以空格隔开的数字，代表全排列数字的起始点 m 和终止点 n 。

保证 $0 \leq m < n \leq 20$ ，同时 $n - m \leq 5$ 。

输出格式

输出有 $(n - m + 1)!$ 行，每行都是从 m 到 n 所有整数的一个排列可能，各整数之间以空格分隔。各行上的全排列不重复。输出按照字典序从小到大排列，即各行遵循“小数优先”原则，在各全排列中，较小的数尽量靠前输出。如果将每行上的输出看成一个数字，则所有输出构成升序数列。有关字典序更详细的解释见HINT。

输入样例1

```
0 2
```

输出样例1

```
0 1 2
0 2 1
1 0 2
1 2 0
2 0 1
2 1 0
```

输入样例2

```
6 9
```

输出样例2

6 7 8 9
6 7 9 8
6 8 7 9
6 8 9 7
6 9 7 8
6 9 8 7
7 6 8 9
7 6 9 8
7 8 6 9
7 8 9 6
7 9 6 8
7 9 8 6
8 6 7 9
8 6 9 7
8 7 6 9
8 7 9 6
8 9 6 7
8 9 7 6
9 6 7 8
9 6 8 7
9 7 6 8
9 7 8 6
9 8 6 7
9 8 7 6

HINT

字典序，顾名思义，就是按照字典的顺序进行排列。根据维基百科的定义：给定两个偏序集 A 和 B ， (a, b) 和 (a', b') 属于笛卡尔集 $A \times B$ ，则字典序定义为 $(a, b) \leq (a', b')$ 当且仅当 $a < a'$ 或 $(a = a' \text{ \& \& } b \leq b')$ 。所以给定两个字符串，逐个字符比较，那么先出现较小字符的那个串字典顺序小，如果字符一直相等，较短的串字典顺序小。例如： $abc < abcd < abde < afab$ 。

总结一下，字典序排序其实就是同一组字符组成的一系列字符串,其规律有：

起点：字典序最小的排列， $1 \sim n$, 例如12345

终点：字典序最大的排列， $n \sim 1$, 例如54321

过程：从当前字符串排列生成字典序刚好比它大的下一个排列，比如12345的字典序下一个排列是12354

NOT HINT

"... ... 吟诗软件其实十分简单，用人类的C语言表达可能超不过两千行代码，另外再加一个存贮所有汉字字符的不大的数据库。当这个软件在位于海王星轨道上的那台量子计算机（一个漂浮在太空中的巨大透明锥体）上启动时，终极吟诗就开始了。... ..."

——[刘慈欣《诗云》](#)

author:czy

I 大师问题

题目描述

引言：曾经，有一位英雄。英雄带领十只宝可梦，向神奥发起了挑战。在那场大战中，人类证明了自己的坚强。神奥认可人的坚强，回到了不属于人世的世界。——《古老诗文10》

梦想成为传说中的英雄的甲贺忍蛙，决定和小智一起挑战宝可梦世界锦标赛的大师。

宝可梦联盟的地砖是 n 乘 n 黑白方格形的， n 位大师每人需要站在一个格子上。大师可以挑战水平、垂直、斜向共八个方向的对手。

称大师之间处于和平共处状态为，谁也不会挑战到其他大师。

问，有多少种不同的和平站法？

输入

第一行为一个数 n ，范围不超过 10。

输出

可以和平共处的站法数。

输入样例

8

输出样例

92

Hint

题目原型：八皇后问题。

后记：追随古代英雄的十只宝可梦，将责任留给子子孙孙，那是代代相传的承诺。人们来到战场，摆下供奉的器皿，那是对宝可梦的报答。鲜采的美食，清冽的净水，那是无尽谢意的象征。——《古老诗文4》



Author：想成为英雄的呱呱泡蛙

J dwy逛超市

题目描述

西方不能没有耶路撒冷，就像 AAUB 不能没有 超市，但作为一个神奇的超市，超市 只接受和找零特定面值的货币，请你帮助不会数学的 dwy 判断她能否使用特定面值的货币在 超市 购买总价值为 k 的商品。

假设 dwy 和 超市 都有足够多数量的货币。

输入格式

共 $2t + 1$ 行。

第一行，一个整数 t ，表示共有 t 组数据。

接下来的 $2t$ 行，每 2 行为一组数据。每组数据的第一行为两个整数，分别表示货币种数 n 和商品的总价值 k ，第二行有 n 个数据，表示每种货币的面值。

输出格式

共 t 行。

对于每组数据输出一行，若可以购买，输出 Da!，否则输出 Nyet.

输入样例

```
3
1 2
1
2 4
3 6
3 5
1 6 9
```

输出样例

```
Da!
Nyet.
Da!
```

样例解释

对于第一组数据，商品总价值为 2，超市只接受面值为 1 的货币，`dwy` 可以使用 2 张面值为 1 的货币完成购买。

对于第二组数据，`dwy` 无法完成购买。

对于第三组数据，商品总价值为 5，超市只接受面值为 1，6 和 9 的货币，`dwy` 可以使用 1 张面值为 6 的货币，超市找零 1 张面值为 1 的货币，完成购买。

数据范围

$$1 \leq n \leq 100$$

$$1 \leq n_i \leq 100$$

$$0 \leq k \leq 10^5$$

数据保证货币的面值各不相同。

HINT

Author: dwy