【题目描述】第1题

给定一个整数数组 nums 和一个整数目标值 target，请你在该数组中找出和为目标值 target 的那**两个**整数，并按照升序返回它们的数组下标。

每种输入**有且仅有**一个答案。但是数组中同一个元素在答案里不能重复出现。或者说数组有重复的元素，但是同一个元素不能用两次。

例如，数组为{2，2} 可以用2+2 但是数组为{2} 不能用2+2

【输入格式】

一行输入n代表数组大小（ 0 < n < 1000）

一行输入n个数代表数组元素

一行输入target

【输出格式】

一行两个数表示数字下标，用空格分隔

【输入样例1】

4

2 7 9 11

9

【输出样例1】

0 1

【输入样例2】

10

0 1 2 3 4 5 6 7 8 9

17

【输出样例2】

8 9

【题目描述】第2题

所谓角谷猜想，是指对于任意一个正整数，如果是奇数，则乘3加1，如果是偶数，则除以2，得到的结果再按照上述规则重复处理，最终总能够得到1。如，假定初始整数为5，计算过程分别为16、8、4、2、1。

程序要求输入一个整数，将经过处理得到1的过程输出来。

【输入格式】

一个正整数N代表要处理的数字

【输出格式】

从输入整数到1的步骤，每一步为一行，每一部中描述计算过程。最后一行输出"End"。如果输入为1，直接输出"End"。

【输入样例】

5

【输出样例】

5\*3+1=16

16/2=8

8/2=4

4/2=2

2/2=1

End

【题目描述】第3题

现在有n块积木，编号从1到n。

一开始，将n块积木按照顺序从左往右摆成一排。

接下来，进行m次操作，每次把位置b上的积木移到位置a的积木的**上面，其余积木不动**。（a等于b则不需要移动任何积木）

4

例如：1234567 将4处积木移动到5，变成123 567

（4在5的上边，且位置4处没有积木）

【输入格式】

第一行为两个正整数n，m (n < 100, m < 1000)

接下来m行，每行输入两个正整数a，b

【输出格式】

输出n行，第i行代表第i个位置**从上到下**的积木编号，若该位置没有积木，则输出一行空行

【输入样例】

2 2

1 2

1 2

【输出样例】

1 2

（此处应有一行空行）

【题目描述】第4题

雇佣兵的体力最大值为M，初始体力值为0、战斗力为N、拥有X个能量元素。

当雇佣兵的体力值恰好为M时，才可以参加一个为期M天的战斗期，战斗期结束体力值将为0。在同一个战斗期内，雇佣兵每连续战斗n天，战斗力就会上升1点，n为当前战斗期开始时的战斗力。

一个战斗期结束后，雇佣兵需要用若干个能量元素使其体力恢复到最大值M，从而参加下一个战斗期。每个能量元素恢复的体力值不超过当前的战斗力。每个能量元素只能使用一次。

请问：雇佣兵的战斗力最大可以到达多少。

【输入格式】

一行包括三个整数M、N、X，相邻两个整数之间用单个空格隔开。

【输出格式】

输出一个整数，为雇佣兵的最大战斗力。

【输入样例】

5 2 10

【输出样例】

6

【输出解释】

|  |  |  |  |
| --- | --- | --- | --- |
| 雇佣兵体力值 | 雇佣兵战斗力 | 能量元素个数 | 雇佣兵体力最大值 |
| 0 | 2 | 10 | 5 |
| 5 | 2 | 7 | 5 |
| 0 | 4 | 7 | 5 |
| 5 | 4 | 4 | 5 |
| 0 | 5 | 4 | 5 |
| 5 | 5 | 1 | 5 |
| 0 | 6(最大值) | 1 | 5 |

【题目描述】第5题

如果在将所有大写字符转换为小写字符、并移除所有非字母数字字符之后，短语正着读和反着读都一样。则可以认为该短语是一个 回文串 。

字母和数字都属于字母数字字符。空格不属于字母数字字符。

给你一个字符串 s，如果它是 回文串 ，返回 true ；否则，返回 false 。【输入格式】

一行为一个字符串

【输出格式】

输出true或false

【输入样例1】

A man, a pen, one night , a marvel

【输出样例1】

false

【输入样例2】

I love C++ c++ eVOl I

【输出样例2】

true

【题目描述】第6题

在经历了字符串的学习后，想必大家已经被char\*,char\*\*折磨的受不了了，在本道题中，将带领大家掌握另一种字符串类型string的使用

（string是一个类，内部封装了char\*，并且包含一些方法）

（如果出现未知错误，可以尝试#include<string>）

1. 构造函数与赋值操作

string str; //构造一个字符串

string str(n,c); //初始化str为n个字符c

要求：输入一个字符串str，接收并输出该字符串

要求：输入一个字符c以及一个正整数n，输出一个由n个c组成的字符串

1. 字符串、字符的拼接

string str1 = “Happy”;

string str2 = “New Year”;

char c = ‘!’;

cout << str1 + “ ” + str2 + c << endl; //输出Happy New Year!

要求：输入三个字符串以及一个字符，输出三者按顺序拼接的结果并保存

1. 字符串的查找

string str; //str为②中拼接好的字符串

string s; char c;

int indexFront = str.find(s); //返回s在str中第一次出现的下标

int indexBack = str.rfind(c); //返回c在str中最后一次出现的下标

(查找失败函数返回 -1 )

要求：输入一个字符串，查找其第一次出现在②中的下标

要求：输入一个字符，查找其最后一次出现在②中的下标

（查找失败输出“Fail”）

1. 字符串的比较

string str1,str2;

str1 > str2 //按照字符ASCII码大小比较

重新完成 第十周上机—第十三题 **按字母顺序输出城市名清单**

要求：输入：第一行一个整数 n 表示城市名的个数接下来 n 行每行一个字符串表示城市名

输出：1 行 n 个字符串,表示城市名按顺序排序后的结果

1. 字符串和整型数据的转化

#include<sstream>

string str;

int a = stoi(str);

double b = stod(str);

要求：输入两个字符串，输出两个字符串对应数字相乘的结果

(输入、输出均为double类型)

【输入样例】

Hello!

10 h

abcd efg abc k

bc t

3

chongqing

beijing

shanghai

6.2.2.2 33.0%s2 (有效数字：6.2 33.0)

【输出样例】

Hello!

hhhhhhhhhh

abcdefgabck

1

Fail

beijing chongqing shanghai

204.6

【题目描述】第7题

请统计某个给定范围[L, R]的所有整数中，数字2出现的次数。

比如给定范围[2, 22]，数字2在数2中出现了1次，在数12中出现1次，在数20中出现1次，在数21中出现1次，在数22中出现2次，所以数字2在该范围内一共出现了6次。

【输入格式】

一行两个正整数，代表L与R

【输出格式】

输出数字2出现的次数

【输入样例1】

2 22

【输出样例1】

6

【输入样例2】

2 100

【输出样例2】

20

【题目描述】第8题

用二维数组存放和操作矩阵

矩阵是一个数表，下图为一个3×5的矩阵：

1 2 3 4 5

6 7 8 9 0

1 2 3 4 5

【输入格式】

一行两个正整数m,n 代表m×n矩阵(m,n≤100)

m行数据，每行数据n个数，代表矩阵对应位置的值

一行正整数op，代表接下来输入的操作数(op ≤ 10000)

op行数据，每行包括三个正整数x,y,z

若x = 0 交换矩阵的第y行和第z行

若x = 1 交换矩阵的第y列和第z列

【输出格式】

变化后的矩阵

【输入样例】

2 2

1 2

3 4

2

0 1 2

1 1 2

【输出样例】

4 3

2 1

【题目描述】第9题

贝壳校园墙更新了新功能，大家可以自由的发布微博。在一条微博中，可以@很多个人（例如：我程序设计基础期末考试没考好，只拿了一百分@室友1@室友2@室友3，则该条微博@了三个人）。在后台，一条微博会被分解成：发送者编号 + 微博中@的人数 + 被@的人的编号。

如今后台收集到了N条微博，他们想看看谁是贝壳最受欢迎的人，即被@次数最多的人，并获取所有@他的人。

【输入格式】

第1行一个正整数N

接下来N行，每行包含：

发送者序号a，共@人数k，k个被@人的序号b1,b2,...,bk

( 1 < N < 10000 , 0 ≤ k ≤ 20 )

【输出格式】

输出占2行

第一行是被@次数最多的人的序号

第二行是所有@他的人的序号(升序排列，同一个序号只用输出一次)

【输入样例】

5

1 2 3 4

1 0

90 3 1 2 4

4 2 3 2

2 1 3

【输出样例】

3

1 2 4

【题目描述】第10题

给定一个数组，它的第 i 个元素表示一支给定股票第 i 天的价格。你只能选择 某一天 买入这只股票，并选择在 **未来的某一个不同的日子** 卖出该股票。设计一个算法来计算你所能获取的最大利润。返回你可以从这笔交易中获取的最大利润。如果你不能获取任何利润，返回 0 。

注意！！！股票只能买卖一次！！！

【输入格式】

一行一个正整数n，表示数组大小

n个自然数，表示数组的值，用空格分开

【输出格式】

一行一个数字表示能获得的最大利益

【输入样例1】

6

7 1 5 3 6 4

【输出样例1】

5

【输入样例2】

7

7 6 5 4 3 2 1

【输出样例2】

0