

# 2022 CCF 非专业级软件能力认证模拟赛 - 入门组

2022/7/28

(请选手务必仔细阅读本页内容)

题目名称	密码强度	打地鼠	选数	跳跳棋
题目类型	传统型	传统型	传统型	传统型
英文题目名称	password	mouse	number	chess
输入文件名	password.in	mouse.in	number.in	chess.in
输出文件名	password.out	mouse.out	number.out	chess.out
输出文件名	1s	1s	1s	1s
内存上限	512M	512M	512M	512M
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	无	无	有	有
结果比较方式	全文比较	全文比较	全文比较	全文比较

## 注意事项:

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回类型必须是 `int`，程序正常结束时的返回值必须是 0。

# 1 密码强度

(password.cpp/c)

## 【题目描述】

一个好的密码有以下四部分组成：

1. 第一部分仅由小写英文字母组成
2. 第二部分仅由大写英文字母组成
3. 第三部分仅由 0 到 9 的数字组成
4. 第四部分由其余特殊字符组成

现在给你一个密码，我们要验证这个密码的强度，首先要把这四部分依次从密码中筛选出来。筛选出来的四部分要保留它们在原字符串中的相对顺序。

比如将"1q2w3E4R6" 这个字符串进行筛选后四部分分别为："qw"、"ER"、"123456"、"{"。

密码的强度最低 1 级，最高 4 级，有几部分不为空，则密码强度就是几级。请你输出这个密码的强度，以及每一部分的内容。

## 【输入输出格式】

### 【输入格式】

从文件 password.in 中读入数据。

仅一行一个字符串 s，表示给你的密码。

### 【输出格式】

输出到文件 password.out 中。

首先输出一行 password level:X，X 表示密码等级，最低为 1 级，最高 4 级。

接下来输出 4 行，表示四部分的内容，输出时要注意保留它们在原字符串中的顺序。如果某一部分为空串，则改为在该行输出 (Null)

## 【样例】

### 【样例 1 输入】

123456

### 【样例 1 输出】

password level:1

(Null)

(Null)

123456

(Null)

**【样例 2 输入】**

Pass\_Word

**【样例 2 输出】**

password level:3

assord

PW

(Null)

-

**【数据范围与提示】**

对于 20% 的测试数据，保证仅有小写英文字母组成，且  $1 \leq |s| \leq 100$

对于 40% 的测试数据，保证仅有大小写英文字母组成，且  $1 \leq |s| \leq 100$

对于 100% 的测试数据，保证字符串是不含空格、回车、或者其他不可见字符的非空字符串，且保证字符串长度  $1 \leq |s| \leq 10^4$

## 2 打地鼠

(mouse.cpp/c/pas)

### 【题目描述】

地鼠游戏是一项需要反应速度和敏捷判断力的游戏。

游戏开始时，会在地板上一下子冒出很多地鼠来，然后等你用榔头去敲击这些地鼠，每个地鼠被敲击后，将会增加相应的游戏分值。

问题是这些地鼠不会傻傻地等你去敲击，它总会在冒出一会时间后又钻到地板下面去（而且再也不上来），每个地鼠冒出后停留的时间可能是不同的，而且每个地鼠被敲击后增加的游戏分值也可能是不同，为了胜出，游戏参与者就必须根据每个地鼠的特性，有选择地尽快敲击一些地鼠，使得总的得分最大。你每敲击一下都要花费 1 单位的时间。

现在告诉你每个地鼠冒出来后停留的时间以及每个地鼠被敲击后将会增加的分值。请你计算出打地鼠游戏能获得的最大分值。

### 【输入输出格式】

#### 【输入格式】

输入文件 mouse.in 包含 3 行，第一行包含一个整数  $n$ ，表示有  $n$  个地鼠从地上冒出来，第二行  $n$  个用空格分隔的整数表示每个地鼠冒出后停留的时间，第三行  $n$  个用空格分隔的整数表示每个地鼠被敲击后会增加的分值）。

每行中第  $i$  个数都表示第  $i$  个地鼠的信息。

#### 【输出格式】

输出文件 mouse.out 只有一行一个整数，表示计算出的最大总分值。

### 【样例】

#### 【样例 1 输入】

```
5
5 3 6 1 4
7 9 2 1 5
```

#### 【样例 1 输出】

```
24
```

### 【数据范围与提示】

对于 30% 的数据：  $1 \leq n \leq 10$

对于 60% 的数据:  $1 \leq n \leq 40$

对于 100% 的数据:  $1 \leq n \leq 100$

地鼠的分值和时间均不超过 1000

### 3 选数

(number.cpp/c/pas)

#### 【题目描述】

给你  $n$  个数，你要选出尽可能多的数，同时满足任意两个数的乘积开三次方根不会是一个整数。

#### 【输入输出格式】

##### 【输入格式】

从文件 number.in 中读入数据。

第一行输入一个正整数  $n$

接下来输入一行  $n$  个正整数  $a_i$

##### 【输出格式】

输出到文件 number.out

输出一行一个正整数，表示最多可以选出多少数，你不需要输出选的方案。

#### 【样例】

##### 【样例 1 输入】

4

4 2 16 27

##### 【样例 1 输出】

3

显然 2 和 4 不能同时选  $2 \times 4 = 2^3$ , 4 和 16 也不能同时选  $4 \times 16 = 4^3$ 。所以你可以选择 2, 16, 27 是符合要求的。

#### 【数据范围与提示】

对于 10% 的测试数据，保证  $1 \leq n \leq 10, 1 \leq a_i \leq 500$

对于 20% 的测试数据，保证  $1 \leq n \leq 10, 1 \leq a_i \leq 10^9$

对于 30% 的测试数据，保证  $1 \leq n \leq 150, 1 \leq a_i \leq 2 \times 10^9$

对于 40% 的测试数据，保证  $1 \leq n \leq 1000, 1 \leq a_i \leq 2 \times 10^9$

对于 100% 的测试数据，保证  $1 \leq n \leq 10^5, 1 \leq a_i \leq 2 \times 10^9$

## 4 跳跳棋

(chess.cpp/c/pas)

### 【题目描述】

陈老师有一种神秘的棋叫做跳跳棋，跳跳棋棋盘可以看成是一个一维的线性数组，编号从 1 到  $n+1$ 。

一开始陈老师的棋子位于第 1 个格子，游戏的最终目的是将棋子移动到第  $n+1$  个格子。

棋盘 1  $n$  的每个格子都有一个“弹力系数”的权值  $p_i$  当棋子位于第  $i$  个格子时，它的下一步可以移动到  $[i - p_i, i + p_i]$  范围内的任意一个格子。

举例来说，假设第 3 个格子的弹力系数为 2，那么陈老师下一步可以移动到第 1,2,3,4,5 格中的任意一格。现在给定 1 到  $n$  每格的弹力系数  $p_i$ ，陈老师发现，好像有时由于棋盘的  $p_i$  置不合理，导致无法通关。所以陈老师化身大法师准备施展神奇的魔法，他每次施展魔法都可以使得一个格子的弹力系数  $p_i + 1$ ，他可以施展若干次魔法操作不同的格子，但是要求陈老师不能够重复对一个格子施展魔法。

陈老师想要知道，为了使跳跳棋能够通关，他最少施展多少次魔法，并且他应该操作哪些格子。

请输出陈老师的最小操作次数，以及施展魔法的操作序列，操作序列的第  $i$  个数表示该次施展魔法的格子编号，由于答案不唯一，所以请你输出一个最小字典序的答案。最小字典序指：在保证第 1 个数字尽可能小的前提下，保证第 2 个数字尽可能的小，依次类推。

### 【输入输出格式】

#### 【输入格式】

从文件 chess.in 中读入数据

第一行输入一个正整数  $n$  表示跳跳棋的格子数目。

接下来输入一行  $n$  个非负整数  $p_i$  表示跳跳棋前  $1 - n$  个格子的弹力系数

#### 【输出格式】

输出到文件 chess.out 中

首先输出一个非负整数 ans，表示少施展魔法的次数。

如果 ans 不为 0，则再输出一行 ans 个整数表示需要施展魔法的格子编号，请给出一个最小字典序的答案。

### 【样例】

#### 【样例 1 输入】

12

5 4 3 3 2 1 0 0 0 1 0 0

**【样例 1 输出】**

5

4 8 9 10 12

除了”4 8 9 10 12”这个操作的答案序列以外，”5 8 9 10 12”，”6 8 9 10 12”也同样是最小操作数下的答案。但是”4 8 9 10 12”这个答案是字典序最小的，故输出”4 8 9 10 12”。

**【样例 1 输入】**

8

0 1 0 1 0 1 0 1

**【样例 1 输出】**

4

1 2 4 6

**【数据范围与提示】**

对于 20% 的测试数据，保证  $1 \leq n \leq 10$

对于 40% 的测试数据，保证  $0 \leq p_i \leq 1$

对于 100% 的测试数据，保证  $1 \leq n \leq 10^5, 0 \leq p_i \leq 100$