

# 1 阅卷 (task.cpp/task.in/task.out/1s/256MB)

## 1.1 题目描述

CZY 是一个中雅培粹学校的初二学生。

众所周知，初中生的大型考试一般都是填答题卡，计算机扫描，然后自动检查选择题并判分，并把后面的题的答案截取下来给老师判分。

又是一场大型考试。考试过后，学校突然发现选择题阅卷系统出了问题。

然而售后服务跑路了，学校的老师又不会写阅卷系统，于是就让 CZY 解决这个问题。

CZY 本来准备重写一份阅卷系统的，突然想到，先看一眼是什么问题，直接打补丁就好啊。

一看：小题分和正确答案的存储没有问题，扫描也没有问题，就是计算分数和判断正误的部分出了点问题。

CZY 觉得这个太简单了，于是就交给了你。

## 1.2 输入格式

第一行一个整数  $n$ ，表示有  $n$  道题。

接下来一行  $n$  个数，第  $i$  个数表示第  $i$  题的小题分。

接下来一个长度为  $n$  的字符串，表示标准答案。

接下来一个长度为  $n$  的字符串，表示这个学生的答案。

## 1.3 输出格式

第一行一个整数  $s$ ，表示这个学生在选择题部分上共获得  $s$  分。

接下来  $n$  行每行一个字符串，如果这题选对了输出 “Accepted”，否则输出 “Wrong Answer”

## 1.4 样例输入 1

```
5
1 1 1 1 1
ABCDA
CCCCC
```

## 1.5 样例输出 1

```
1
Wrong Answer
Wrong Answer
Accepted
Wrong Answer
Wrong Answer
```

## 1.6 说明与提示

选择题每题选项只可能为  $ABCD$  之一。

$n \leq 60$ (初中英语通常有 60 个选择题当然以后会只有 59 个)

每个题的分数  $\leq 5$

## 2 匹配 (match.cpp/match.in/match.out/1s/256MB)

### 2.1 题目描述

CZY 是一个中雅培粹学校的初二学生。

他经常有将一个长字符串压缩的习惯。

这个压缩，就是把字符串中一些没用的删掉，然后对于连续的一段被删除的字符，用一个 \* 代替。

比如一个字符串：qwertyuiopasdfghjkl，删除一些字符后：(x 表示被删除) qxerxxuiopaxxxhijkl，然后就会变成 q\*er\*uiopa\*hijkl

在这之后，如果串尾有 \*，CZY 可能会把 \* 删掉。

在删除后，CZY 会顺手把原串删掉。

由于这些没用的东西实在是没用，所以这个新串在大多数时候都是够了的。

然而有一次 CZY 需要用到原串后发现原串被删除了，于是拼命回忆，想到了一个字符串  $S$ 。

他想知道  $S$  能不能压缩成这个字符串  $T$ ，如果不可能，那这个串一定不是原串。如果能，那这个串可能是原串。

### 2.2 输入格式

第一行一个整数  $t$  表示有  $t$  组数据。

接下来  $2t$  行，每两行表示一组数据。

对于每组数据，第一行一个字符串  $S$ ，第二行一个字符串  $T$ 。

### 2.3 输出格式

对于每组数据，如果可能输出 "Maybe"，否则输出 "Impossible"

### 2.4 样例输入

```
3
zhongyapeicui
zh*ieicui
zhongyapeicui
zh*eicui
zhongyapeicui
zhongya*peicui
```

### 2.5 样例输出

```
Impossible
Maybe
Impossible
```

### 2.6 数据范围

$t \leq 10, |S|, |T| \leq 1000$

保证  $S$  仅由小写字母构成， $T$  仅由小写字母和 \* 构成。

保证不会出现连续多于一个 \*

### 3 数学题 (math.cpp/math.in/math.out/1s/256MB)

#### 3.1 题目描述

CZY 是一个中雅培粹学校的初二学生。

今天他学了一个重要的知识点：整除，约数和倍数。

他立刻想到一个问题：给定  $x$ ，求  $x$  的约数个数。

他想了一下，把这题秒了。但是他想考考你，于是把这题扔给了你。

请你帮他解决这个问题。

#### 3.2 输入格式

一行一个数  $x$

#### 3.3 输出格式

一个数  $ans$  表示有  $ans$  个数是  $x$  的约数。

#### 3.4 样例输入

15

#### 3.5 样例输出

4

#### 3.6 样例解释

1, 3, 5, 15 都是 15 的约数。

#### 3.7 提示

50%: $x \leq 10^5$

另 20%: $x$  是质数

100%: $x \leq 10^9$

你可能会用到：

对于任意  $a|b, a \leq \sqrt{b}$  和  $\frac{b}{a} \leq \sqrt{b}$  至少有一个成立

## 4 食物 (food.cpp/food.in/food.out/1s/256MB)

### 4.1 题目描述

CZY 是一个中雅培粹学校的初二学生，很喜欢吃东西。

有  $n$  种食物，第  $i$  种食物会在  $start_i$  时间出炉，然后就能吃了。吃完它需要  $time_i$  的时间。

CZY 不能同时吃两种食物。

CZY 初始（时间 0 时）饥饿度为 0，每单位时间若不在吃东西，饥饿度都会 +1。

当 CZY 吃东西的时候，他的饥饿度会变成 0。

饥饿程度为在所有东西吃完之前，CZY 的饥饿度的最大值。

CZY 希望饥饿程度最小。

求：最小的饥饿程度。

### 4.2 输入格式

第一行一个整数  $n$ ，表示食物的种类。

接下来  $n$  行，每行两个整数  $start_i, time_i$  表示第  $i$  样食物出炉时间和食用需要时间。

### 4.3 输出格式

最小的饥饿程度。

### 4.4 样例输入

```
3
1 3
5 6
10 1
```

### 4.5 样例输出

```
1
```

### 4.6 样例解释

第 0 秒等待，饥饿值 1。

第 1 秒到第 3 秒，吃完了食物 1，饥饿值 0

第 4 秒等待，饥饿值 1

第 5 秒到第 11 秒，吃完食物 2，饥饿值 0，在第 10 秒食物 3 出炉。

第 12 秒吃完食物 3，饥饿值 0，饥饿程度为 1，可以证明没有饥饿程度更小的方案。

### 4.7 数据范围

30%:  $start_i \leq 10, time_i = 0, n \leq 5$

60%:  $start_i, time_i \leq 1000, n \leq 100$

另 20%:  $start_i = 0, n \leq 1000$

100%:  $start_i, time_i \leq 10^9, n \leq 1000$