

第十三届蓝桥杯大赛软件赛决赛

C/C++ 大学 B 组

【考生须知】

考试开始后，选手首先下载题目，并使用考场现场公布的解压密码解压试题。

考试时间为 4 小时。考试期间选手可浏览自己已经提交的答案，被浏览的答案允许拷贝。时间截止后，将无法继续提交或浏览答案。

对同一题目，选手可多次提交答案，以最后一次提交的答案为准。

选手必须通过浏览器方式提交自己的答案。选手在其它位置的作答或其它方式提交的答案无效。

试题包含“结果填空”和“程序设计”两种题型。

结果填空题：要求选手根据题目描述直接填写结果。求解方式不限。不求源代码。把结果填空的答案直接通过网页提交即可，不要书写多余的内容。

程序设计题：要求选手设计的程序对于给定的输入能给出正确的输出结果。考生的程序只有能运行出正确结果才有机会得分。

注意：在评卷时使用的输入数据与试卷中给出的示例数据可能是不同的。选手的程序必须是通用的，不能只对试卷中给定的数据有效。

对于编程题目，要求选手给出的解答完全符合 GNU C/C++ 标准，不能使用诸如绘图、Win32API、中断调用、硬件操作或与操作系统相关的 API。

代码中允许使用 STL 类库。

注意：main 函数结束必须返回 0。

注意：所有依赖的函数必须明确地在源文件中 `#include <xxx>`，不能通过工程设置而省略常用头文件。

所有源码必须在同一文件中。调试通过后，拷贝提交。

提交时，注意选择所期望的编译器类型。

试题 A: 2022

本题总分：5 分

【问题描述】

将 2022 拆分成 10 个互不相同的正整数之和，总共有多少种拆分方法？

注意交换顺序视为同一种方法，例如 $2022 = 1000 + 1022$ 和 $2022 = 1022 + 1000$ 就视为同一种方法。

【答案提交】

这是一道结果填空的题，你只需要算出结果后提交即可。本题的结果为一个整数，在提交答案时只填写这个整数，填写多余的内容将无法得分。

试题 B: 钟表

本题总分：5 分

【问题描述】

在 12 小时制的钟表中，有分针、时针、秒针来表示时间。记分针和时针之间的夹角度数为 A ($0 \leq A \leq 180$)、分针和秒针之间的夹角度数为 B ($0 \leq B \leq 180$)。而恰好在 s 时 f 分 m 秒时，满足条件 $A = 2B$ 且 $0 \leq s \leq 6$; $0 \leq f < 60$; $0 \leq m < 60$ ，请问 s, f, m 分别是多少。

注意时针、分针、秒针都围绕中心匀速转动。

提交格式为三个由一个空格隔开的整数，分别表示 s, f, m 。如 **3 11 58** 表示 3 点 11 分 58 秒。

【答案提交】

这是一道结果填空的题，你只需要算出结果后提交即可。本题的结果为三个由一个空格隔开的整数，在提交答案时只填写为三个由一个空格隔开的整数，填写多余的内容将无法得分。

试题 C: 卡牌

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 10 分

【问题描述】

这天, 小明在整理他的卡牌。

他一共有 n 种卡牌, 第 i 种卡牌上印有正整数数 $i(i \in [1, n])$, 且第 i 种卡牌现有 a_i 张。

而如果有 n 张卡牌, 其中每种卡牌各一张, 那么这 n 张卡牌可以被称为一套牌。小明为了凑出尽可能多套牌, 拿出了 m 张空白牌, 他可以在上面写上数 i , 将其当做第 i 种牌来凑出套牌。然而小明觉得手写的牌不太美观, 决定第 i 种牌最多手写 b_i 张。

请问小明最多能凑出多少套牌?

【输入格式】

输入共 3 行, 第一行为两个正整数 n, m 。

第二行为 n 个正整数 a_1, a_2, \dots, a_n 。

第三行为 n 个正整数 b_1, b_2, \dots, b_n 。

【输出格式】

一行, 一个整数表示答案。

【样例输入】

```
4 5
1 2 3 4
5 5 5 5
```

【样例输出】

```
3
```

【样例说明】

这 5 张空白牌中，拿 2 张写 1，拿 1 张写 2，这样每种牌的牌数就变为了 3,3,3,4，可以凑出 3 套牌，剩下 2 张空白牌不能再帮助小明凑出一套。

【评测用例规模与约定】

对于 30% 的数据，保证 $n \leq 2000$ ；

对于 100% 的数据，保证 $n \leq 2 \times 10^5$ ； $a_i, b_i \leq n$ ； $m \leq n^2$ 。

试题 D: 最大数字

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 10 分

【问题描述】

给定一个正整数 N 。你可以对 N 的任意一位数字执行任意次以下 2 种操作:

1. 将该位数字加 1。如果该位数字已经是 9，加 1 之后变成 0。
2. 将该位数字减 1。如果该位数字已经是 0，减 1 之后变成 9。

你现在总共可以执行 1 号操作不超过 A 次，2 号操作不超过 B 次。

请问你最大可以将 N 变成多少？

【输入格式】

第一行包含 3 个整数: N, A, B 。

【输出格式】

一个整数代表答案。

【样例输入】

123 1 2

【样例输出】

933

【样例说明】

对百位数字执行 2 次 2 号操作，对十位数字执行 1 次 1 号操作。

【评测用例规模与约定】

对于 30% 的数据， $1 \leq N \leq 100$; $0 \leq A, B \leq 10$

对于 100% 的数据， $1 \leq N \leq 10^{17}$; $0 \leq A, B \leq 100$

试题 E: 出差

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 15 分

【问题描述】

A 国有 N 个城市, 编号为 $1 \dots N$ 。小明是编号为 1 的城市中一家公司的员工, 今天突然接到了上级通知需要去编号为 N 的城市出差。

由于疫情原因, 很多直达的交通方式暂时关闭, 小明无法乘坐飞机直接从城市 1 到达城市 N , 需要通过其他城市进行陆路交通中转。小明通过交通信息网, 查询到了 M 条城市之间仍然还开通的路线信息以及每一条路线需要花费的时间。

同样由于疫情原因, 小明到达一个城市后需要隔离观察一段时间才能离开该城市前往其他城市。通过网络, 小明也查询到了各个城市的隔离信息。(由于小明之前在城市 1, 因此可以直接离开城市 1, 不需要隔离)

由于上级要求, 小明希望能够尽快赶到城市 N , 因此他求助于你, 希望你能帮他规划一条路线, 能够在最短时间内到达城市 N 。

【输入格式】

第 1 行: 两个正整数 N, M , N 表示 A 国的城市数量, M 表示未关闭的路线数量

第 2 行: N 个正整数, 第 i 个整数 C_i 表示到达编号为 i 的城市后需要隔离的时间

第 $3 \dots M+2$ 行: 每行 3 个正整数, u, v, c , 表示有一条城市 u 到城市 v 的双向路线仍然开通着, 通过该路线的时间为 c

【输出格式】

第 1 行: 1 个正整数, 表示小明从城市 1 出发到达城市 N 的最短时间 (到达城市 N , 不需要计算城市 N 的隔离时间)

【样例输入】

```
4 4
5 7 3 4
1 2 4
1 3 5
2 4 3
3 4 5
```

【样例输出】

```
13
```

【样例说明】

```
[1 (5)] --4-- [2 (7)]
|             |
|             |
5             3
|             |
|             |
[3 (3)] --5-- [4 (4)]
```

路线 1: 1 -> 2 -> 4, 时间为 $4+7(\text{隔离})+3=14$

路线 2: 1 -> 3 -> 4, 时间为 $5+3(\text{隔离})+5=13$

【评测用例规模与约定】

对于 100% 的数据, $1 \leq N \leq 1000$, $1 \leq M \leq 10000$, $1 \leq C_i \leq 200$, $1 \leq u, v \leq N$, $1 \leq c \leq 1000$

试题 F: 费用报销

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 15 分

【问题描述】

小明在出差结束后返回了公司所在的城市, 在填写差旅报销申请时, 粗心的小明发现自己弄丢了出差过程中的票据。

为了弥补小明的损失, 公司同意小明用别的票据进行报销, 但是公司财务要求小明提交的票据中任意两张的日期差不小于 K 天, 且总金额不得超过实际差旅费用 M 。

比如财务要求 $K = 7$ 时, 若小明提交了一张 1 月 8 日的票据, 小明就不能提交 1 月 2 日至 1 月 14 日之间的其他票据, 1 月 1 日及之前和 1 月 15 日及之后的票据则可以提交。

公司的同事们一起给小明凑了 N 张票据, 小明现在想要请你帮他整理一下, 从中选取出符合财务要求的票据, 并使总金额尽可能接近 M 。

需要注意, 由于这些票据都是同一年的, 因此 12 月底的票据不会影响到 1 月初票据的提交。这一年不是闰年。

【输入格式】

第 1 行: 3 个整数, N, M, K

第 2... $N+1$ 行: 每行 3 个整数 m_i, d_i, v_i , 第 $i+1$ 行表示第 i 张票据时间的月份 m_i 和日期 d_i , v_i 表示该票据的面值

【输出格式】

第 1 行: 1 个整数, 表示小明能够凑出的最大报销金额

【样例输入】

```
4 16 3
1 1 1
```

1 3 2

1 4 4

1 6 8

【样例输出】

10

【样例说明】

选择 1 月 3 日和 1 月 6 日的票据

【评测用例规模与约定】

对于 100% 的评测用例， $1 \leq N \leq 1000$, $1 \leq M \leq 5000$, $1 \leq K \leq 50$, $1 \leq m_i \leq 12$, $1 \leq d_i \leq 31$, $1 \leq v_i \leq 400$

日期保证合法。

试题 G: 故障

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 20 分

【问题描述】

在软件或系统开发中，我们会遇到各种各样的故障。为了从故障现象反推故障原因，工程师们会总结一种叫做相关性矩阵的二维表格，来表示故障原因与故障现象之间的关系。比如：

		1	2	3	4	5
A		x	x	x		
B	x		x			
C				x	x	

其中每行表示一种故障原因，每一列表示一种故障现象。该矩阵表示故障原因 A 可能产生故障现象 2、3、4，故障原因 B 可能产生故障现象 1、3。

在实际开发过程中，如果出现了故障原因，工程师就可以根据故障现象，去计算每种故障原因产生的概率，并按照概率大小对故障原因进行排查，以达到快速定位故障原因的目的。

现在，我们假设系统开发中同一时间只会出现一种故障原因，并且故障原因引起各故障现象是独立事件。举个例子来说：

假设系统现在发生了故障原因 A，有 $\frac{1}{3}$ 的概率出现故障现象 2，有 $\frac{1}{4}$ 的概率出现故障现象 3，有 $\frac{1}{2}$ 的概率出现故障现象 4。由于 3 种现象是独立发生的，因此有 $\frac{1}{2 \times 3 \times 4}$ 的概率同时出现故障 2、3、4。

约定若相关性矩阵中没有 ‘x’ 记号，则表示该故障原因一定不会产生某故障现象，比如故障原因 A，一定不会产生故障现象 1。

根据历史经验数据，我们统计得到了每一种故障原因出现的概率以及每一种故障原因对应的故障现象产生概率。

现在已知系统出现的故障现象，求问各个故障原因发生的概率。

【输入格式】

第 1 行：2 个正整数 N, M ， N 表示故障原因的个数（编号 $1 \dots N$ ）， M 表示故障现象的个数（编号 $1 \dots M$ ）

第 2 行： N 个整数，第 i 个数表示故障原因 i 产生的概率 P_i 。

第 $3 \dots N+2$ 行：每行 $M+1$ 个整数，第 $i+1$ 行第 j 个整数 P_{ij} 表示故障原因 i 出现故障现象 j 的概率（百分比）。

第 $N+3$ 行：1 个正整数 K ，表示目前出现的故障现象数量

第 $N+4$ 行： K 个正整数，依次为当前出现的故障现象编号，不会重复

【输出格式】

第 $1 \dots N$ 行：按概率从高到低输出每种故障原因及其可能的概率，若出现概率相同则优先输出编号小的故障原因。第 1 个数为故障原因编号，第 2 个数为故障概率（百分比），保留 2 位小数。

【样例输入】

```
3 5
30 20 50
0 50 33 25 0
30 0 35 0 0
0 0 0 25 60
1
3
```

【样例输出】

```
2 56.89
1 43.11
3 0.00
```

【评测用例规模与约定】

对于所有测试用例， $1 \leq N \leq 40$, $1 \leq M \leq 20$, $0 \leq P_i \leq 100$, $\sum(P_i) = 100$,
 $0 \leq P_{ij} \leq 100$

试题 H: 机房

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 20 分

【问题描述】

这天, 小明在机房学习。

他发现机房里一共有 n 台电脑, 编号为 1 到 n , 电脑和电脑之间有网线连接, 一共有 $n - 1$ 根网线将 n 台电脑连接起来使得任意两台电脑都直接或者间接地相连。

小明发现每台电脑转发、发送或者接受信息需要的时间取决于这台电脑和多少台电脑直接相连, 而信息在网线中的传播时间可以忽略。比如如果某台电脑用网线直接连接了另外 d 台电脑, 那么任何经过这台电脑的信息都会延迟 d 单位时间 (发送方和接收方也会产生这样的延迟, 当然如果发送方和接收方都是同一台电脑就只会产生一次延迟)。

小明一共产生了 m 个疑问: 如果电脑 u_i 向电脑 v_i 发送信息, 那么信息从 u_i 传到 v_i 的最短时间是多少?

【输入格式】

输入共 $n + m$ 行, 第一行为两个正整数 n, m 。

后面 $n - 1$ 行, 每行两个正整数 x, y 表示编号为 x 和 y 的两台电脑用网线直接相连。

后面 m 行, 每行两个正整数 u_i, v_i 表示小明的第 i 个疑问。

【输出格式】

输出共 m 行, 第 i 行一个正整数表示小明第 i 个疑问的答案。

【样例输入】

```
4 3
1 2
```

1 3
2 4
2 3
3 4
3 3

【样例输出】

5
6
1

【样例说明】

这四台电脑各自的延迟分别为 2, 2, 1, 1。

对于第一个询问，从 2 到 3 需要经过 2, 1, 3，所以时间和为 $2 + 2 + 1 = 5$ 。

对于第二个询问，从 3 到 4 需要经过 3, 1, 2, 4，所以时间和为 $1 + 2 + 2 + 1 = 6$ 。

对于第三个询问，从 3 到 3 只会产生一次延迟，所以时间为 1。

【评测用例规模与约定】

对于 30% 的数据，保证 $n, m \leq 1000$ ；

对于 100% 的数据，保证 $n, m \leq 100000$ 。

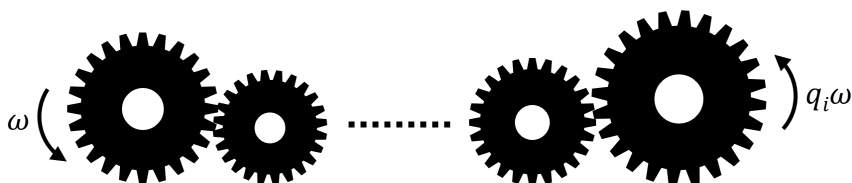
试题 I: 齿轮

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 25 分

【问题描述】

这天, 小明在组装齿轮。

他一共有 n 个齿轮, 第 i 个齿轮的半径为 r_i , 他需要把这 n 个齿轮按一定顺序从左到右组装起来, 这样最左边的齿轮转起来之后, 可以传递到最右边的齿轮, 并且这些齿轮能够起到提升或者降低转速 (角速度) 的作用。



小明看着这些齿轮, 突然有 Q 个疑问: 能否按一定顺序组装这些齿轮使得最右边的齿轮的转速是最左边的齿轮的 q_i 倍?

【输入格式】

输入共 $Q + 2$ 行, 第一行为两个正整数 n, Q , 表示齿轮数量和询问数量。

第二行为 n 个正整数 r_1, r_2, \dots, r_n , 表示每个齿轮的半径。

后面 Q 行, 每行一个正整数 q_i 表示询问。

【输出格式】

Q 行, 对于每个询问, 如果存在至少一种组装方案满足条件, 输出 ‘YES’, 否则输出 ‘NO’。

【样例输入】

5 3

4 2 3 3 1

2

4

6

【样例输出】

YES

YES

NO

【样例说明】

询问 1 方案之一：2 3 3 4 1

询问 2 方案之一：4 2 3 3 1

询问 3 没有方案

【评测用例规模与约定】

对于 15% 的数据，保证 $n, Q \leq 100$ ；

对于 30% 的数据，保证 $n, Q \leq 2000$ ；

对于 100% 的数据，保证 $n, Q \leq 2 \times 10^5$; $a_i, q_i \leq 2 \times 10^5$ 。

试题 J: 搬砖

时间限制: 1.0s 内存限制: 256.0MB 本题总分: 25 分

【问题描述】

这天, 小明在搬砖。

他一共有 n 块砖, 他发现第 i 砖的重量为 w_i , 价值为 v_i 。他突然想从这些砖中选一些出来从下到上堆成一座塔, 并且对于塔中的每一块砖来说, 它上面所有砖的重量和不能超过它自身的价值。

他想知道这样堆成的塔的总价值 (即塔中所有砖块的价值和) 最大是多少。

【输入格式】

输入共 $n + 1$ 行, 第一行为一个正整数 n , 表示砖块的数量。

后面 n 行, 每行两个正整数 w_i, v_i 分别表示每块砖的重量和价值。

【输出格式】

一行, 一个整数表示答案。

【样例输入】

```
5
4 4
1 1
5 2
5 5
4 3
```

【样例输出】

```
10
```

【样例说明】

选择第 1、2、4 块砖，从上到下按照 2、1、4 的顺序堆成一座塔，总价值为 $4 + 1 + 5 = 10$

【评测用例规模与约定】

对于 20% 的数据，保证 $n \leq 10$ ；

对于 100% 的数据，保证 $n \leq 1000$; $w_i \leq 20$; $v_i \leq 20000$ 。