2026 第五届昆明青少年人工智能综合实践大赛

C++编程决赛（预备组）

考试时间：2026 年 4 月 25 日 14：00-16：00

题目名称	时间旅行	采购水果	三角洲行动	宝藏
题目类型	传统型	传统型	传统型	传统型
目录	time	fruit	force	treasure
可执行文件名	time	fruit	force	treasure
输入文件名	time.in	fruit.in	force.in	treasure.in
输出文件名	time.out	fruit.out	force.out	treasure.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	128MB	128MB	256MB	256MB
子任务数目	10	10	10	20
测试点是否等分	是	是	是	是

提交源程序文件名

对于 C++ 语言	time.cpp	fruit.cpp	force.cpp	treasure.cpp
对于 C 语言	time.c	fruit.c	force.c	treasure.c

编译选项

对于 C++ 语言	-O2 -lm
对于 C 语言	-O2 -lm

注意事项（请仔细阅读）

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() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考文件提交格式的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 全省统一评测时采用的机器配置为：Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz 16G 内存。上述时限以此配置为准。
7. 只提供 Linux 格式附加样例文件。
8. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

时间旅行（time）

【题目描述】

小明有一个电子表，显示当前时间（时、分、秒，24 小时制）。他想要知道经过若干秒后，表上显示的时间是多少。注意：时间为 24 时制，如果超过 24 时，则从 0 时重新开始（例如 23:59:59 经过 1 秒后变为 00:00:00）。

请你编写程序，输入当前时间的时、分、秒，以及一个整数秒数（非负），输出经过后的时间（时、分、秒）。

【输入格式】

从文件 **time.in** 中读入数据。

一行四个正整数h, m, s, t，分别表示当前时、分、秒和经过的秒数。

【输出格式】

输出到文件 **time.out** 中。

输出一行三个整数，表示新的时、分、秒，用空格隔开。

【样例 1 输入】

```
14 30 45 1000
```

【样例 1 输出】

```
14 47 25
```

【样例 1 解释】

输入当前时间 14:30:45，经过 1000 秒后，通过时间进制换算得到新的时间为 14:47:25，未跨天。

【样例 2 输入】

```
23 59 59 1
```

【样例 2 输出】

```
0 0 0
```

【样例 2 解释】

输入当前时间 23:59:59，经过 1 秒后，总秒数恰好达到 86400（一天的秒数），因此时间归零为 00:00:00，实现跨天。

【数据范围】

对于所有测试数据保证：$0 \le h \le 23, 0 \le m, s \le 59, 0 \le t \le 10^7$。

采购水果（fruit）

【题目描述】

小向是一家大型连锁超市的采购经理，他计划在 4-5 月份采购水果来销售。

目前市场上新鲜的水果有两种：一种是樱桃，售价a元每千克，另外一种是羊奶果，售价b元每千克。

小向的采购经费为c元，他计划采购若干整数千克的樱桃和羊奶果。

小向想知道在正好花完c元经费的情况下，有多少种不同的采购方案。

【输入格式】

从文件 **fruit.in** 中读入数据。本题包含多组测试数据。

第一行一个数字t，表示测试数据的组数。

第 2 行到第$t+1$行，每行表示一组测试数据，包含五个数字a、b、c，分别表示樱桃的单价、羊奶果的单价、采购的经费。

【输出格式】

输出到文件 **fruit.out** 中。

输出包含t行，每行 1 个数字，表示每组测试数据的采购方案数。

【样例 1 输入】

```
2
2 3 21
4 4 16
```

【样例 1 输出】

```
4
5
```

【样例 1 解释】

有两组测试数据。

第一组樱桃是 2 元每千克，羊奶果是 3 元每千克，经费是 21 元。有 4 种采购方案，分别是：

樱桃 0 千克，羊奶果 7 千克；樱桃 3 千克，羊奶果 5 千克；樱桃 6 千克，羊奶果 3 千克；樱桃 9 千克，羊奶果 1 千克。

第二组樱桃是 4 元每千克，羊奶果是 4 元每千克，经费是 16 元。有 5 种采购方案，分别是：

樱桃 0 千克，羊奶果 4 千克；樱桃 1 千克，羊奶果 3 千克；樱桃 2 千克，羊奶果 2 千克；樱桃 3 千克，羊奶果 1 千克；樱桃 4 千克，羊奶果 1 千克。

【样例 2】

见选手目录下 fruit / fruit2.in 和 fruit / fruit2.ans。

【样例 3】

见选手目录下 fruit / fruit3.in 和 fruit / fruit3.ans。

【数据范围】

对于所有测试数据有：$1 \le t \le 100, 1 \le a, b \le 10^5, 1 \le c \le 10^{18}$。

测试点	$a, b \le$	$c \le$	特殊性质
1	10	10^2	无
2~3	10^3	10^6	无
4	10^3	10^9	$a = 1$
5	10^3	10^{18}	无
6	10^5	10^9	a、b互质
7~8	10^5	10^9	无
9~10	10^5	10^{18}	无

三角洲行动（force）

【题目背景】

三角洲是一款多平台跨端第一人称特战干员战术射击游戏，现在各个年龄段都非常流行，小李也是其中玩家之一。

【题目描述】

在一局游戏中，小李准备完成一次“危险行动”模式中的机密任务。小李需要在他的仓库中的n个武器、药品等物品中，选择若干物品携带进入这局游戏，其中第i个物品有这些属性：

1.占用背包的格数w_i

2.系统给出物品的战备值b_i

3.小李对于物品的战斗力评价v_i

根据任务要求及小李的风险偏好，小李希望在物品占用背包格数不超过W，总战备值不超过B时，尽可能高的战斗力总和进入这局游戏。

小李对这个问题很头疼，于是他找到了你，他希望你告诉他选择哪些物品进入游戏以及最大战斗力总和。

【输入格式】

从文件 **force.in** 中读入数据。

第一行包含三个数字n, W, B,分别表示物品的数量、背包的格数上限、总战备值上限。

第二行到$n+1$行，每行三个数字w_i、b_i、v_i，分别表示第i个物品占用背包的格数、战备值、战斗力。

【输出格式】

输出到文件 **force.out** 中。

第一行仅一个数字，为最大战斗力总和。

第二行输出若干个物品的编号，不同物品编号之间用一个空格分割，为选择

的物品的编号列表，顺序任意。如果有多组方案的战斗力总和均为最大值，则以任意顺序输出任意一组均可。

注意：因为本题包含 Special Judge，请不要输出多余的空格，否则可能会导致判定出错。

【样例 1 输入】

```
4 20 20
7 8 9
3 10 6
9 9 4
4 5 5
```

【样例 1 输出】

```
15
2 1
```

【样例 1 解释】

总共有 4 种物品，物品的属性见下表。物品占用格数不得超过 20，总战备值不得超过 20。

物品编号	占用背包的格数	战备值	战斗力
1	7	8	9
2	3	10	6
3	9	9	4
4	4	5	5

选择 1、2 物品最优，占用背包的总格数为 7+3=10，总战备值为 8+10=18，总战斗力为 9+6=15。不存在比这个方案更优的方案。

【样例 2】

见选手目录下 force / force2.in 和 force / force2.ans。

【样例 3】

见选手目录下 force / force2.in 和 force / force3.ans。

【数据范围】

对于所有测试数据保证：$1 \le n \le 100, 1 \le W, B \le 200$，$1 \le w_i \le W$，$1 \le b_i \le B, 1 \le v_i \le 200$。

测试点	n ≤	特殊性质
1~2	10	无
3~5	15	无
6	100	$w_i = b_i = 1$
7~8	100	$w_i = 1$
9~10	100	无

如果只输出了正确的最大战斗力总和，但是没有输出选择的物品列表、输出物品的列表不符合题意，每个测试点均可以获得一半的分数。如果输出了错误的最大战斗力总和，不论物品列表是否正确，都不得分。

宝藏（treasure）

【题目描述】

小 A 在历经一系列冒险之后终于得知了宝藏的方位。在经过调查之后，小 A 在地图上标注了自己的位置和宝藏的位置，当然，沿途充满着危险，一些位置无法经过，并且其余位置每走一步都需要消耗掉一定量的体力和花费一定量的时间。

在体力为 0 时无法移动，万幸的是，一些位置有神奇的草药并且可以无限采摘，吃下就会立即回复一定量的体力。

小 A 想知道自己最快多久可以到达宝藏的所在地。

给定一个大小为 $n \times m$ 的矩阵地图，一个起点坐标为 (sx, sy)，一个终点坐标为 (ex, ey)。从起点出发，每次可以选择往上下左右四个方向移动一格，每次移动花费 1 单位时间以及 1 单位体力。小 A 的初始体力大小和体力上限为 k，当体力减小为 0 之后无法再移动。

地图上的每个格子用一个字符表示，格子有下面三种情况：

. ：表示这个格子可以移动。

：表示这个格子为障碍物，无法移动。

一个个位数字 x，（$1 \le x \le 9$）：表示移动到这个格子之后，会恢复 x 点体力，如果移动到该格子体力为 0，那么也会恢复 x 点体力。注意恢复体力无法超过体力的上限。每次经过都可以恢复体力，到达终点时的体力可以为 0。

保证起点和终点的位置一定为 . 。

【输入格式】

从文件 **treasure.in** 中读入数据。

输入的第一行包括 7 个整数 n, m, sx, sy, ex, ey, k，分别表示地图行数和列数、起点、终点、初始体力及最大体力。

接下来 n 行，每行 m 个字符，表示矩阵地图每个格子的信息。

【输出格式】

输出到文件 **treasure.out** 中。

输出仅 1 个数字，如果能够到达，则输出最快到达的时间；如果永远不可能到达，输出 -1 。

【样例 1 输入】

```
3 10 2 2 2 8 5
#####5####
#.....#
##########
```

【样例 1 输出】

```
8
```

【样例 1 解释】

在样例中，起点为 (2,2)，终点为 (2,8)，唯一的方案为从起点一直走到 (1,6)，吃完草药再直接前往终点，一共需要 8 步。

【样例 2】

见选手目录下 treasure / treasure2.in 和 treasure / treasure3.ans。

【样例 3】

见选手目录下 treasure / treasure3.in 和 treasure / treasure3.ans。

【样例 4】

见选手目录下 treasure / treasure4.in 和 treasure / treasure4.ans。

【数据范围】

对于所有测试数据保证：$1 \le n, m \le 1000$，$1 \le k \le 10$。

测试点	n, $m \le$	$k \le$
1~3	10	10
5	1000	1
6~20	1000	10