第四届（2025）青少年编程挑战

精英赛入门组试卷

考试时间：2025 年 10 月 12 日

题目名称	数字分割	旋转舞台	分蛋糕	挖迷宫
题目类型	传统型	传统型	传统型	传统型
目录	digital	stage	cake	laby
可执行文件名	digital	stage	cake	laby
输入文件名	digital.in	stage.in	cake.in	laby.in
输出文件名	digital.out	stage.out	cake.out	laby.out
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	1.0 秒
内存限制	512MB	512MB	512MB	512MB
子任务数目	10	10	20	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	digital.cpp	stage.cpp	cake.cpp	laby.cpp
对于 C 语言	digital.c	stage.c	cake.c	laby.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。

数字分割（digital）

【题目描述】

给定一个正整数num，将num分割为两个非负整数，将分割的两个数拼接起来得到num各位数的任意一个排列，且出现的数字次数与num中数字出现次数保持一致（分离出的两个数可以包含前导 0）。

请你编程计算出分割出来的两个数的和最小是多少？分割出来的两个数需要满足以下规律：

①为保证数字和的位数尽可能小，两个数字的位数要尽可能的接近。

②为保证两个数字各自尽可能小，两个数字的最高位要尽可能小，且保证最高位的情况下再使得次高位尽可能小……以此类推。

【输入格式】

从文件 **digital.in** 中读入数据。

输入一行一个正整数num。

【输出格式】

输出到文件 **digital.out** 中。

输出分割后两个数的最小和。

【样例 1 输入】

```
94760
```

【样例 1 输出】

```
116
```

【样例 1 解释】

当 94760 分割为 069 和 47 时两个数的和最小，和为 116。

【样例 2 输入】

```
9811
```

【样例 2 输出】

```
37
```

【样例 2 解释】

将 9811 分割为 18 和 19 两个数，和最小，和为 37。

【样例 3】

见选手目录下的 digital/ digital3.in 和 digital / digital3.ans。

【数据范围】

对于所有测试数据保证：$1 \leq num \leq 10^9$。

旋转舞台（stage）

【题目描述】

奇幻舞台剧下个月就要公演了，小项需要对舞台进行布置，营造出亦真亦幻的效果。小项把舞台剧分隔成n行m列的网格，在每个网格内放上颜色各异的砖块用来搭阶梯。在他的想法中，把舞台背景旋转起来后，观众有着场景阶梯上升的感觉。

小项把他手头的砖块按照颜色的深度，把砖块从1到$n \times m$编号，将第1块砖放置到第x行y列的网格中，之后，按如下方式放置编号为K（$K = 2,3,4,\ldots,n \times m$）的砖块：

①小项根据$K-1$编号砖块位置的上一行右一列，暂定K编号砖块的位置。

②如果暂定位置的行号是0，将行号改为n。

③如果暂定位置的行号是$n+1$，将行号改为1。

④如果暂定位置的列号是0，将列号改为m。

⑤如果暂定位置的列号是$m+1$，将列号改为1。

⑥如果暂定位置上已经确定了砖块的编号，那么改为$K-1$编号砖块的位置的下一行，然后再检查步骤②~步骤⑤。

小项的舞台很大，他不希望你输出所有的数据，他目前只希望知道p个位置的砖块编号，以方便他施工。

【输入格式】

从文件 **stage.in** 中读入数据。

输入第一行包含五个数字n、m、p、x、y。分别表示舞台的行数、舞台的列数、小项关心的位置个数、第一个砖块所在的行号、第一个砖块所在的列号。

接着输入p行，每行两个数字a、b，表示小项想要知道的砖块所在的a行号和b列号。

【输出格式】

输出到文件 **stage.out** 中。

输出p行，每行一个数字，表示网格砖块编号。

【样例1输入】

```
3 3 3 1 2
1 3
2 2
3 3
```

【样例 1 输出】

```
6
5
2
```

【样例 1 解释】

根据规则，每个网格放置的砖块编号如下：

8	1	6
3	5	7
4	9	2

【样例 2 输入】

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

【样例 2 输出】

```
17
16
7
14
```

【样例 2 解释】

根据规则，每个网格放置的砖块编号如下：

5	1	17	13	9
20	16	12	8	4
15	11	7	3	19
10	6	2	18	14

【样例 3】

见选手目录下 stage/ stage3.in 和 stage / stage3.ans。

【数据范围】

对于所有测试数据有：$1 \leq n, m \leq 1000, 1 \leq a, x \leq n, 1 \leq b, y \leq m$。

测试点	$n, m \leq$
1~2	10
3~6	100
7~10	1000

分蛋糕（cake）

【题目描述】

小项和朋友们一共k个人，刚刚购买了一个长度为m，宽度为n的矩形蛋糕，准备大伙一起分着吃，不过这个蛋糕上面的草莓不是很均匀。如果我们把这个矩形蛋糕划分成$n \times m$的网格，可以观察得第i行第j列的网格上有$a_{i,j}$颗草莓。

大家都因为草莓分配的事情争论不休，每个人都想获得更多的草莓，于是，聪明的小项想出了一个好办法，就由他自己来负责切蛋糕，不过他只能拿到最后一块蛋糕（最后一块蛋糕草莓最少）。

特别需要说明的是，小项每次只能从一块蛋糕的边缘，沿着每个网格的边缘线横切或者竖切蛋糕直到蛋糕的另外一个边缘，并且不能改变刀的移动方向从而把一块蛋糕分成两块，求出小项把整个蛋糕分成k块的所有方案中，他能获得最多草莓的数量。

【输入格式】

从文件 **cake.in** 中读入数据。

第一行包含三个数字n, m, k，分别表示蛋糕的长度、蛋糕的宽度、及蛋糕分成的块数。

接下来n行，每行m个数字，每个数字$a_{i,j}$表示每个网格上的草莓数量。

【输出格式】

输出到文件 **cake.out** 中。

输出仅一个数字，表示小项能够获得的最大草莓数。

【样例 1 输入】

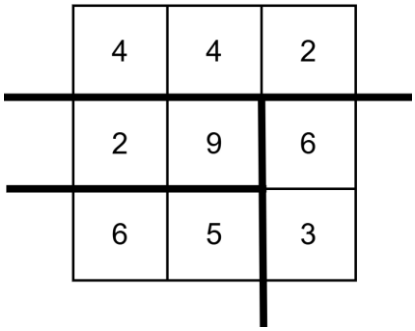
```
3 3 4
4 4 2
2 9 6
6 5 3
```

【样例 1 输出】

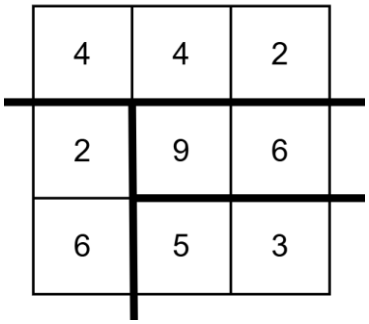
```
9
```

【样例 1 解释】

如下图切割蛋糕的方案最优（加粗黑线为切割处），4 块蛋糕的草莓数分别是 10、11、11、9。小项是最后一个取蛋糕的人，只能分得最少的草莓 9。



下图也是一种分蛋糕的方案，但是小项只能获得最小的那块，只能获得 8 颗草莓。



【样例 2】

见选手目录下 cake/cake2.in 和 cake/cake2.ans

【样例 3】

见选手目录下 cake/cake3.in 和 cake/cake3.ans

【数据范围】

对于所有测试数据有：$1 \leq n, m \leq 30$，$0 < k \leq n \times m, 0 \leq a_{i,j} \leq 100$

测试点	$n, m \leq$	特殊性质
1~3	5	无
4	5	$k = n \times m$
5~7	10	无
8	10	$k = 1$
9~12	20	无
13	20	$k = n \times m$

14~15	30	所有$a_{i,j}$相同
16~17	30	$n = 1$
18~20	30	无

挖迷宫（laby）

【题目描述】

勇者们马上要攻击作为魔王的你，但你的地下迷宫还没有挖好，且只有一个迷宫的入口。

迷宫由n个洞穴组成，其中 1 号洞穴为迷宫的入口，由$n-1$条道路连通整个迷宫中的所有洞穴，这些道路还未挖掘。不过这些道路有长有短，对于连接洞穴u_i、v_i之间的道路，还需要r_i天才能挖通。每天可以选择一个未挖通的道路挖掘，且此道路连接着某个已挖通的洞穴，在已挖通的迷宫中移动的时间忽略不计（洞穴是天然存在的，不用花时间去挖掘）。

另外为了能够及时获得洞穴中的魔力以对付勇者，你还计划需要在第d_i天前(包括第d_i天)抵达第i个洞穴。现在是第 1 天，你想知道你的计划能否实现。

【输入格式】

从文件 **laby.in** 中读入数据。

每个测试点包含多组测试数据。

第一行包含 1 个数字t, 表示测试数据的组数。

每组测试数据的第一行包含一个整数n，表示洞穴的数量。

接下来一行包含n个数字，第i个数字表示，第i个洞穴最晚需要在第d_i天抵达。保证第一个数字为 0。

接下来$n-1$行，每行包含 3 个数字u_i、v_i、r_i。表示第i条道路连接u_i、v_i两个洞穴，这条道路需要挖r_i天。

【输出格式】

输出到文件 **laby.out** 中。

输出 t 行，每行输出 “Yes”或者“No”，表示能否及时抵达各个洞穴。

【样例 1 输入】

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

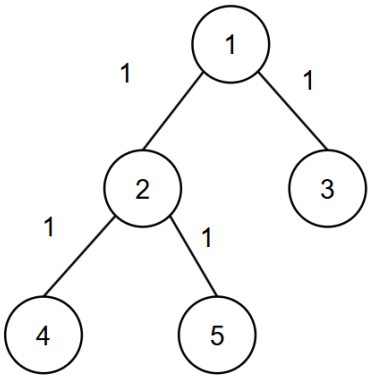
```
1 3 1
2 4 1
4 5 1
```

【样例 1 输出】

```
Yes
```

【样例 1 解释】

样例包含 1 组测试数据。迷宫情况入下图，每条道路均需要花 1 天的时间挖掘，第 2 个洞穴需要在第 1 天之前抵达，第 3 个洞穴需要在第 2 天之前抵达，第 4 个洞穴需要在第 3 天之前抵达，第 5 个洞穴需要在第 4 天之前抵达。



能够及时抵达的方案只有一种。

第一天挖掘洞穴 1 和洞穴 2 之间的道路。正好抵达洞穴 2，洞穴 1、2 已挖通。

第二天挖掘洞穴 1 和洞穴 3 之间的道路。正好在第 2 天抵达洞穴 3，洞穴 1、2、3 已挖通。

第三天挖掘洞穴 2 和洞穴 4 之间的道路。正好在第 3 天抵达洞穴 4，洞穴 1、2、3、4 已挖通。

第四天挖掘洞穴 2 和洞穴 3 之间的道路。正好在第 4 天抵达洞穴 5。

所有洞穴都能在计划时间内抵达。

【样例 2】

见选手目录下的 laby / laby2.in 和 laby / laby2.ans。

【样例 3】

见选手目录下的 laby / laby3.in 和 laby / laby3.ans。

【数据范围】

对于所有测试数据有：$1 \le t \le 10, 1 \le n \le 10^5, 0 \le r_i \le 10^9, 1 \le d_i \le 10^9$

测试点	t ≤	n ≤	特殊性质
1~2	10	10^2	A
3~5		10^3	无
6~7		10^3	B
8~10		10^4	无
11		10^4	D
12~13		10^4	C
14		5×10^4	A
15		5×10^4	C
16		5×10^4	D
17~20		5×10^4	无

特殊性质 A：$r_i = 1$

特殊性质 B：所有道路的两个洞穴编号均满足，$u_i = v_i + 1$

特殊性质 C：所有洞穴的所需抵达时间d_i均相同，$d_i = d_j$。

特殊性质 D：所有道路的其中一个洞穴均为洞穴 1，$u_i = 1$。