

全锅青少年信息学奥林匹克联赛模拟赛

NOIP 模拟赛

时间：2024 年 9 月 28 日 7: 50 ~ 11: 50

题目名称	a	b	c	d
题目类型是否为传统型	1	1	1	1
目录	a	b	c	d
可执行文件名	a	b	c	d
输入文件名	a.in	b.in	c.in	d.in
输出文件名	a.out	b.out	c.out	d.out
每个测试点时限	1.0 秒	1.5 秒	1.0 秒	0.5 秒
内存限制	512 MB	512 MB	1024 MB	1024 MB
子任务数目	4	4	4	5
子任务是否等分	0	0	0	0

提交源程序文件名

对于 C++ 语言	a.cpp	b.cpp	c.cpp	d.cpp
-----------	-------	-------	-------	-------

编译选项

对于 C++ 语言	-std=c++14 -lm -O2
-----------	--------------------

注意事项（请仔细阅读）

1. 由于宇宙射线，题目不一定按难度顺序排列。T3 的难度有可能小于 T2。
2. 所有题目均采用捆绑测试。
3. 若数据出错由验题人负责，若数据太弱由出题人赛后加强数据。

$$a(a)$$
【题目描述】

定义对序列 a 进行一次变换是指同时将 a_i 变为 a_i 在 $a_1, a_2, \dots, a_{i-1}, a_i$ 中出现的次数。

请你求出将长度为 n 的序列 a 进行 m 次变换后的结果。

【输入格式】

从文件 ***a.in*** 中读入数据。

第一行包含两个整数 n, m 。

第二行包含 n 个整数，即为序列 a 。

【输出格式】

输出到文件 ***a.out*** 中。

输出一行，包含 n 个用空格分隔的整数，表示 m 次变换后的序列 a 。

【样例 1 输入】

```
1 10 1
2 1 3 2 4 1 3 1 4 2 1
```

【样例 1 输出】

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

【样例 2 输入】

```
1 10 2
2 1 3 2 4 1 3 1 4 2 1
```

【样例 2 输出】

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

【样例 3】

见选手目录下的 ***a/a3.in*** 与 ***a/a3.ans***。

【数据范围】

对于 100% 的数据， $1 \leq a_i \leq n \leq 10^6, 1 \leq m \leq 10^9$ 。

各测试点的附加限制如下表所示：

测试点编号	$n \leq$	$m \leq$	特殊性质	分值
1	5000	5000		20
2	10^6	10^9	$a_i \leq 2$	20
3	9	10^9		20
4	10^6	10^9		40

$$b(b)$$
【题目描述】

tyr 有 $3n$ 张牌，其中每张牌上都写着 $1 \sim n$ 中的一个数，他会重复以下操作 $n - 1$ 次：将最左侧的 5 张牌任意排列后，删去最左侧的 3 张牌，如果这 3 张牌上写着同样的数，tyr 可以获得 1 分。

最后，如果剩余的 3 张牌上的数字一样，那么他还可以额外得到 1 分。

现在，tyr 想知道，他得到的分数最高是多少。

【输入格式】

从文件 *b.in* 中读入数据。

输入两行。

第一行，一个整数 n ，表示牌有 $3n$ 张。

第二行， $3n$ 个整数 a_1, a_2, \dots, a_{3n} ，其中第 i 个数 a_i 表示从左到右第 i 张牌写的数字。

【输出格式】

输出到文件 *b.out* 中。

输出一行一个整数表示答案。

【样例 1 输入】

```
1 2
2 1 2 1 2 2 1
```

【样例 1 输出】

```
1 2
```

【样例 2 输入】

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

【样例 2 输出】

```
1 1
```

【样例 3 输入】

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

【样例 3 输出】

```
1 3
```

【样例 4】

见选手目录下的 *b/b4.in* 与 *b/b4.ans*。

【数据范围】

对于所有数据，保证 $1 \leq a_i \leq n \leq 3 \times 10^3$ 。
各测试点的附加限制如下表所示：

测试点编号	$n \leq$	$a_i \leq$	分值
1	50	n	19
2	3×10^3	2	19
3	3×10^3	3	19
4	3×10^3	n	43

$$c(c)$$

【题目描述】

定义一个矩阵合法当且仅当其中不存在两个位置数值相等。

给定一个 $n \times m$ 的正整数矩阵 a ，求最大的合法的连续子矩阵大小。

【输入格式】

从文件 *c.in* 中读入数据。

第一行输入包含两个整数 n, m 表示矩阵的长宽。

接下来第 1 行至第 n 行每行包含 m 个整数，第 i 行第 j 列的数表示 $a_{i,j}$ 。

【输出格式】

输出到文件 *c.out* 中。

一行一个正整数表示最大子矩阵大小。

【样例 1 输入】

```
1 3 3
2 1 3 1
3 4 5 6
4 2 6 1
```

【样例 1 输出】

```
1 6
```

【样例 2 输入】

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

【样例 2 输出】

```
1 4
```

【样例 3 输入】

```
1 2 6
2 1 2 3 4 5 6
3 8 6 7 8 9 1
```

【样例 3 输出】

```
1 8
```

【样例 4】

见选手目录下的 *c/c4.in* 与 *c/c4.ans*。

【数据范围】

对于 100% 的数据， $1 \leq n, m \leq 400, 1 \leq a_i \leq nm$ 。
各测试点的附加限制如下表所示：

测试点编号	$n, m \leq$	$a_i \leq$	分值
1	40	nm	18
2	100	nm	18
3	400	400	18
4	400	nm	46

$$d(d)$$
【题目描述】

对于一个排列 a ，定义其权值如下：

生成一个 $(n+2) \times (n+2)$ 的网格图，行列标号为 $0 \sim n+1$ ，每次可以从 (i, j) 走到 $(i, j+1)$ 或 $(i+1, j)$ ，且不能走到 (i, a_i) ，权值为从 $(0, 0)$ 走到 $(n+1, n+1)$ 的方案数。

现在排列 a 某些位置被改成了 -1 ，请求出所有可能的初始排列的权值之和 $\text{mod } 998244353$ 。

【输入格式】

从文件 ***d.in*** 中读入数据。

输入两行。

第一行一个整数 n 表示排列长度。

第二行 n 个整数 a_1, a_2, \dots, a_n 表示这个排列。

【输出格式】

输出到文件 ***d.out*** 中。

输出共一行表示你的答案。

【样例 1 输入】

```
1 4
2 1 -1 4 -1
```

【样例 1 输出】

```
1 41
```

【样例 2 输入】

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

【样例 2 输出】

```
1 2
```


【样例 3 输入】

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

【样例 3 输出】

```
1 48
```

【样例 4】

见选手目录下的 *d/d4.in* 与 *d/d4.ans*。

【数据范围】

对于所有数据，保证 $1 \leq n \leq 200$; $1 \leq a_i \leq n$ 或 $a_i = -1$; 若 $i \neq j, a_i \neq -1, a_j \neq -1$ 则 $a_i \neq a_j$ 。
各测试点的附加限制如下表所示：

测试点编号	$n \leq$	特殊性质	分值
1	8		17
2	50		17
3	200	$1 \leq a_i$	17
4	200	$a_i = -1$	17
5	200		32