

NOIP 模拟赛

2022.10

题目名称	密码锁	距离	不等	选取
题目类型	传统	传统	传统	传统
提交文件名	lock.cpp	dist.cpp	neq.cpp	choose.cpp
输入文件名	lock.in	dist.in	neq.in	choose.in
输出文件名	lock.out	dist.out	neq.out	choose.out
时间限制	1s	2s	5s	2s
内存限制	512MB	512MB	512MB	1GB
测试点数目	20	20	20	20
测试点是否等分	是	是	是	是
是否有附加文件	是	是	是	是

Problem A. 密码锁 (lock.cpp)

Input file: lock.in
Output file: lock.out
Time limit: 1 second
Memory limit: 512 megabytes

你有一个 N 位数的密码锁。

为了解开这个密码锁，你要对密码锁进行若干次操作，每次操作可以选取密码锁上任意、**连续**的一段数位并将它们同时向上/向下拨动**恰好一位**。

以 $N = 4$ 为例，假设密码锁的当前状态是 0492。在一次操作中，你可以选取 049 这段数位并将它们向上拨动得到 9382，也可以将它们向下拨动得到 1502。

给出 Q 次询问，每次询问给出密码锁的初始状态与你想要到达的最终状态，求从初始状态至最终状态所需操作数的最小值。

Input

从文件 **lock.in** 中读入数据。

第一行，一个正整数 N ，保证 $1 \leq N \leq 4$ 。

第二行，一个正整数 Q 表示询问的数量，保证 $1 \leq Q \leq 100000$ 。

接下来的 Q 行，每行两个长度为 N 的十进制数字串，依次表示密码锁的初始状态与最终状态。

Output

输出到文件 **lock.out** 中。

输出一共 Q 行，每行一个非负整数表示对应询问的答案。

Examples

lock.in	lock.out
2	1
5	2
12 23	4
12 24	5
87 91	0
56 11	
12 12	

题面中的样例对应选手目录下的 **lock/lock1.in** 与 **lock/lock1.ans**。

样例 2 满足 $N = 3$ ， $Q = 1000$ ，见选手目录下的 **lock/lock2.in** 与 **lock/lock2.ans**。

样例 3 满足 $N = 4$ ， $Q = 10000$ ，见选手目录下的 **lock/lock3.in** 与 **lock/lock3.ans**。

Notes

本题共有 20 个测试点，标号为 $1, 2, \dots, 20$ 。

对于测试点 $1, 2$ ，保证 $N = 1$ 。

对于测试点 $3, 4, 5, 6$ ，保证 $1 \leq N \leq 2$ 。

对于测试点 $7, 8, 9, 10, 11, 12$ ，保证 $1 \leq N \leq 3$ 。

对于所有测试点，保证 $1 \leq N \leq 4$ ， $1 \leq Q \leq 100000$ 。

Problem B. 距离 (dist.cpp)

Input file: `dist.in`
Output file: `dist.out`
Time limit: 2 seconds
Memory limit: 512 megabytes

给定两个长度为 N 的非负整数序列 A_0, A_1, \dots, A_{N-1} 和 B_0, B_1, \dots, B_{N-1} , 求

$$\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \min\{|A_i - A_j|, |B_i - B_j|\},$$

其中 $\min\{a, b\}$ 表示 a 与 b 中的较小值。

Input

从文件 `dist.in` 中读入数据。

第一行一个正整数 N , 保证 $1 \leq N \leq 5 \times 10^5$ 。

第二行 N 个非负整数, 依次为 A_0, A_1, \dots, A_{N-1} 。

第三行 N 个非负整数, 依次为 B_0, B_1, \dots, B_{N-1} 。

保证 $0 \leq A_i, B_i \leq 10^6$ 。

Output

输出到文件 `dist.out` 中。

一行一个整数, 表示答案。

Examples

<code>dist.in</code>	<code>dist.out</code>
3 1 3 2 1 2 3	6

题面中的样例对应选手目录下的 `dist/dist1.in` 与 `dist/dist1.ans`。

样例 2 满足 $N = 2000$, 见选手目录下的 `dist/dist2.in` 与 `dist/dist2.ans`。

样例 3 满足 $N = 5 \times 10^5$, 见选手目录下的 `dist/dist3.in` 与 `dist/dist3.ans`。

Notes

本题共有 20 个测试点, 标号为 $1, 2, \dots, 20$ 。

对于测试点 $1, 2, 3, 4$, 保证 $1 \leq N \leq 2000$ 。

对于测试点 $5, 6, 7, 8, 9, 10$, 保证 $0 \leq A_i, B_i \leq 60$ 。

对于所有测试点, 保证 $1 \leq N \leq 5 \times 10^5$, $0 \leq A_i, B_i \leq 10^6$ 。

Problem C. 不等 (neq.cpp)

Input file: `neq.in`
Output file: `neq.out`
Time limit: 5 seconds
Memory limit: 512 megabytes

给定一个长度为 N 的非负整数序列 A_0, A_1, \dots, A_{N-1} , 你要回答 Q 个询问, 询问形式如下:

给定满足 $0 \leq L < R \leq N$ 的非负整数 L, R , 你要求出 $(0, 1, \dots, N-1)$ 的排列 P_0, P_1, \dots, P_{N-1} 的个数, 满足: 对于 $L \leq i < R$, 必定有 $P_i \neq A_i$ 。

因为满足条件的排列个数可能很多, 你只需要输出答案对 $10^9 + 7$ 取模的结果。

Input

从文件 `neq.in` 中读入数据。

第一行两个正整数 N, Q , 保证 $2 \leq N, Q \leq 3000$ 。

第二行 N 个非负整数, 依次为 A_0, A_1, \dots, A_{N-1} , 保证 $0 \leq A_i < N$ 。

接下来的 Q 行, 每行两个非负整数 L 和 R , 保证 $0 \leq L < R \leq N$ 。

Output

输出到文件 `neq.out` 中。

输出 Q 行, 每行一个非负整数 (对 $10^9 + 7$ 取模), 表示对应询问的答案。

Examples

<code>neq.in</code>	<code>neq.out</code>
3 3	4
0 0 0	2
0 1	0
0 2	
0 3	

$L = 0, R = 2$ 的询问要求 $P_0 \neq 0$ 且 $P_1 \neq 0$, 满足条件的排列有 $(1, 2, 0)$ 和 $(2, 1, 0)$ 共 2 种。

题面中的样例对应选手目录下的 `neq/neq1.in` 与 `neq/neq1.ans`。

样例 2 满足 $N = 100$, 见选手目录下的 `neq/neq2.in` 与 `neq/neq2.ans`。

Notes

本题共有 20 个测试点, 标号为 $1, 2, \dots, 20$ 。

对于测试点 $1, 2, 3$, 保证 $2 \leq N, Q \leq 10$ 。

对于测试点 $4, 5, 6, 7, 8$, 保证 $2 \leq N, Q \leq 100$ 。

对于测试点 $9, 10$, 保证 A 中的元素两两不同, 即 A 是一个 $(0, 1, \dots, N-1)$ 的排列。

对于所有测试点，保证 $2 \leq N, Q \leq 3000$, $0 \leq A_i < N$ 。

Problem D. 选取 (choose.cpp)

Input file: choose.in

Output file: choose.out

Time limit: 2 seconds

Memory limit: 1 gigabyte

给定一个长度为 N 的正整数序列 A_0, A_1, \dots, A_{N-1} 。你想要选取其中的一些数，并使选出的数的和最大化。你的选取方式需要满足如下条件：

- 如果 A_i 被选取，且 A_i 左侧连续的 c_i 个数 (c_i 由你的选取情况决定) 也被选取，即 $A_{i-c_i}, A_{i-c_i+1}, \dots, A_{i-1}$ 均被选取但 A_{i-c_i-1} 不被选取 (或是 $i - c_i - 1 < 0$)，那么 c_i 必须满足 $0 \leq c_i < 4$ 。
- 在上述前提下，对于被选取的 A_i ，定义其**代价**为 X_{c_i} ，其中 X_0, X_1, X_2, X_3 是题目给定的长度为 4 的正整数序列。一种选取方式的**总代价**定义为所有被选中的 A_i 对应的 X_{c_i} 之和，那么你的总代价不能超过 T ，其中 T 也是题目给定的正整数。

为了更好地解释选取方式的限制，我们以 $N = 10$, $T = 7$, $X = [1, 1, 1, 2]$ 为例：

- 选取 $A_0, A_1, A_2, A_3, A_5, A_6$ 是**合法**的，其总代价 $(X_0 + X_1 + X_2 + X_3) + (X_0 + X_1) = 7 \leq T$ 。
- 选取 A_1, A_2, A_3, A_4, A_5 是**非法**的，因为 A_5 左侧连续的 4 个数均被选取了。
- 选取 $A_0, A_1, A_2, A_4, A_5, A_6, A_8, A_9$ 是**非法**的，因为其总代价 $8 > T$ 。

你只要求出所有合法的选取方案中选出的 A_i 之和的最大值。

Input

从文件 **choose.in** 中读入数据。

第一行两个正整数 N, T ，保证 $4 \leq N \leq 250$, $1 \leq T \leq 10^9$ 。

第二行 4 个正整数，依次为 X_0, X_1, X_2, X_3 ，保证 $1 \leq X_i \leq 10^9$ 。

第三行 N 个正整数，依次为 A_0, A_1, \dots, A_{N-1} ，保证 $0 \leq A_i \leq 10^9$ 。

Output

输出到文件 **choose.out** 中。

输出一行一个非负整数，表示所求最大值。

Examples

choose.in	choose.out
6 11 1 2 4 8 1 10 100 1000 10000 100000	111011
10 808995774 227150190 224559951 139457454 87915609 96204862 799843967 446173607 796619138 402690754 223219513 668171337 312183499 905549873 673542337	2559447046

题面中的两个样例对应选手目录下的 `choose/choose1.in` 与 `choose/choose1.ans`、`choose/-choose2.in` 与 `choose/choose2.ans`。

样例 3 满足 $N = 100$ ，见选手目录下的 `choose/choose3.in` 与 `choose/choose3.ans`。

Notes

- 本题共有 20 个测试点，标号为 $1, 2, \dots, 20$ ：
- 对于测试点 1, 2，保证 $4 \leq N \leq 15$ 。
- 对于测试点 3, 4, 5, 6，保证 $1 \leq T \leq 10000$ 。
- 对于测试点 7, 8, 9, 10，保证 $4 \leq N \leq 80$ 。
- 对于测试点 11, 12，保证 $4 \leq N \leq 100$ 。
- 对于测试点 13, 14，保证 $4 \leq N \leq 200$ 。
- 对于所有测试点，保证 $4 \leq N \leq 250$ ， $1 \leq T \leq 10^9$ 。