Problem A. 卡尔的礼物

Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes



卡尔给出两个正整数a, b，你需要找到两个数字c, d，满足:

- $min(c, d) > max(a, b)$;
- $gcd(c, d) = gcd(a, b)$.
- $1 \le c, d \le 10^{15}$

其中$gcd(a, b)$为a, b的最大公因数。
可能存在多组满足题意的答案，你只要输出一组可行的解即可，显然满足限制条件的解是恒存在的。

Input

第一行是一个整数T $(1 \le T \le 10^6)$,表示样例的个数。 以后每行一个样例，为两个正整数a, b $(1 \le a, b \le 10^9)$。

Output

每行输出一个样例的结果，代表满足限制条件下的一组解c, d。

Example

standard input	standard output
2 2 4 7 8	6 14 100 111

Note

注意:巨大的输入量，推荐使用C风格的输入输出。

Problem B. DUEL!

Input file:	standard input
Output file:	standard output
Time limit:	1.5 seconds
Memory limit:	256 megabytes

某天rabbit正在尝试构建自己的卡组，这样他就能快乐的和别人决斗了。

rabbit将他的卡片布置为一个$n \times m$，每个位置都有一张牌的矩阵。根据大师规则1，rabbit需要至少40张，至多60张卡片。不过rabbit很懒，所以他每次只执行以下两种操作之一:

1. 设当前剩下的卡牌矩阵为$x \times y$。rabbit选择若干行r $(1 \leq r < x)$，然后移除掉这些行上的所有卡牌，矩阵大小变为$(x - r) \times y$。
2. 设当前剩下的卡牌矩阵为$x \times y$。rabbit选择若干列c $(1 \leq c < y)$，然后移除掉这些列上的所有卡牌，矩阵大小变为$x \times (y - c)$。

现在rabbit想要知道最少需要执行多少次该操作才能让剩下的卡牌能够组成卡组，即$40 \leq x \cdot y \leq 60$。

Input

第一行是一个整数T $(1 \leq T \leq 10^6)$,表示样例的个数。以后每行一个样例，为两个整数n, m, $(1 \leq n, m \leq 10^9)$。

Output

每行输出一个样例的结果，表示rabbit最少需要执行的操作数，若无论如何都不能组成卡组，则输出-1。

Example

standard input	standard output
3 20 20 2 20 2 19	1 0 -1

Note

注意:巨大的输入量，推荐使用C风格的输入输出。

Problem C. 折磨王(easy version)

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

此题为折磨王的easy version，其与hard version 的唯一区别在于给定的区间$[l,r]$始终为$[1,n]$。

rabbit喜欢出一些奇怪的问题来折磨Bbaka，这天rabbit丢给Bbaka一个1到n的排列p，并且允许Bbaka执行任意次以下操作:

- 选择两个不同的数字$i(1 \le i \le n)$，$j(1 \le j \le n)$，然后将$p[i]$与$p[j]$交换位置

定义$f(x)$为将x还原为$\{1,2,...,n\}$所需的最小操作数，其中x为一个1到n的排列。现在rabbit有一个数k，他想知道，让初始序列p变为序列p'，且$f(p')=k$ ，最少需要执行多少次上述的操作。**注意：$f(x)$的计算不受下面交换元素的限制区间条件限制**

这个问题对于神Bbaka而言显然有些太简单了，于是Bbaka很快就给出了正解。而且Bbaka在原来的基础上给出了限制区间$[l,r]$，使得每次操作选择的两个数字i,j中，至少有一个数在这个区间。这下rabbit难被到了，于是它向你求助。

Input

输入分为三行:

第一行为两个数字$n(1 \le n \le 200000)$，$k(0 \le k \le n-1)$。

第二行为两个数字$l,r, 1 \le l \le r \le n$，代表Bbaka给出的限制区间，此处保证$l=1, r=n$。

第三行为n个数字，代表初始序列p，输入保证p为1到n的一个排列。

Output

输出一个整数，代表让初始序列p变为序列p'，$f(p')=k$，且每次操作都符合Bbaka的限制下所需的最小操作数。

若不可能构造出满足题意得序列p'，则输出-1。

Examples

standard input	standard output
4 1 1 4 1 2 3 4	1
4 0 1 4 4 2 3 1	1

Note

对于第一个样例，只需要交换$p[3]$与$p[4]$，使得$p'=\{1,2,4,3\}$即可，显然$f(p')=1$满足题意，且不存在花费操作数更少的构造。

对于第二个样例，只需要交换$p[1]$与$p[4]$，使得$p'=\{1,2,3,4\}$即可，显然$f(p')=0$满足题意，且不存在花费操作数更少的构造。

Problem D. 上分

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

paranoid很喜欢打Codeforces。而你，有了特殊能力，能够看到paranoid接下来n场的加减分情况，设为$\{a_1, a_2, \ldots, a_n\}$。而且你还能任意的变换这些场次的顺序，使之变为$\{b_1, b_2, \ldots, b_n\}$。显然交换场次顺序之后得到的 b_i 是 a_i 的一个排列。(paranoid的心理素质很好，他某场的rate变化不会受到场次顺序变化的影响)。现在你想帮助paranoid上分。

初始时，paranoid准备打2场。假设当前paranoid已经打完了第i场，当同时满足以下三个条件时，paranoid准备打的总场次就会加2。

- 当前场次不是第一场，即 $i \neq 1$
- 该场比赛涨分，即 $b_i > 0$
- 上一场也涨分且该场加分的比上一场要多，即 $b_i > b_{i-1} > 0$

注意:paranoid不一定会打完所有的n场，比如一共有4比赛，他第一场涨了2分，第二场涨了1分，此时不满足增加场次的条件，那么他就只打这两场，后面场次不会参加，哪怕后面两场可能会涨分。

即使他打算打更多的场次，实际最多也只能打n场。

现在请你安排接下来n场的题目顺序，使得paranoid能上最多的分。如果有多种方案，只要输出一种即可。

注：为了方便起见，paranoid每一场的加减分都不同。

Input

第一行是一个整数$T(1 \leq T \leq 10,000)$,表示样例的个数。

以后每两行一个样例，每个样例的第一行为$n(2 \leq n \leq 100,000)$，代表你能够看到接下来n场加分情况。

第二行为n个数字$a_1, a_2, \ldots, a_n$，$-10^9 \leq a_i \leq 10^9$，代表初始时第i场的加分，输入保证每个a_i都是独一无二的。

对于每个的测试样例，满足$\sum_1^T n_i \leq 100,000$。

Output

每行输出一个样例的结果，如样例所示。

对于每个样例，输出n个数字，代表一个a_i的排列。即在这个顺序下按照paranoid打Codeforces的规则，他能够上最多的分。

Example

standard input	standard output
2 4 1 3 4 2 4 1 3 2 -3	1 2 3 4 3 2 1 -3

Note

对于样例的第一个测试点的结果:

paranode打了第1场，涨1分，准备一共打2场；

paranode打了第2场，涨2分，因为这场不是第1场且涨分了且涨的比上一场多，他决定多打2场，即准备一共打4场；

paranode打了第3场，涨3分，因为这场不是第1场且涨分了且涨的比上一场多，故他打算打6场，但是实际打的总场次不能超过4场，所以还是一共打4场；

paranode打了第4场，涨4分，因为这场不是第1场且涨分了且涨的比上一场多，故他打算打8场，但是打的总场次不能超过4场，所以打完这场之后paranoid就会停止下来；

一共加了10分，显然不存在比这加分更多的方案。

Problem E. awa症候群

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

小B非常喜欢颜表情awa，以至于他想要把看到的一切字符串都变为若干个awa。

有一次小B遇到了一个长度为n，仅由小写字母构成的字符串s。小B可以执行以下操作任意次:

- 将该字符串上任意位置的字符修改为任意小B指定的小写字母。

现在小B想知道要使得字符串s含有k个awa至少需要多少次上述操作，但是小B最近忙于换寝室，于是他向你求助。

注意一个位置上的字符最多只能属于一个awa，如修改完成后的字符串为$awawa$，则其只含有一个awa而不是2个

Input

第一行代表两个整数n, k，$3 \le n \le 2000$，$0 \le k \le \lfloor \frac{n}{3} \rfloor$。

第二行是一个长度为n仅由小写字符构成的字符串s

Output

输出一个数字，代表使得字符串s含有k个awa至少需要的操作次数。

Example

standard input	standard output
3 1 bbb	3

Problem F. 折磨王(hard version)

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

此题为折磨王的hard version，其与easy version 的唯一区别在于给定的区间$[l,r]$不固定。

rabbit喜欢出一些奇怪的问题来折磨Bbaka，这天rabbit丢给Bbaka一个1到n的排列p，并且允许Bbaka执行任意次以下操作:

- 选择两个不同的数字$i(1 \le i \le n)$，$j(1 \le j \le n)$，然后将$p[i]$与$p[j]$交换位置

定义$f(x)$为将x还原为$\{1,2,...,n\}$所需的最小操作数，其中x为一个1到n的排列。现在rabbit有一个数k，他想知道，让初始序列p变为序列p'，且$f(p')=k$ ，最少需要执行多少次上述的操作。**注意:$f(x)$的计算不受下面交换元素区间的限制**

这个问题对于神Bbaka而言显然有些太简单了，于是Bbaka很快就给出了正解。而且Bbaka在原来的基础上给出了限制区间$[l,r]$，使得每次操作选择的两个数字i,j中，至少有一个数在这个区间。这下rabbit难被到了，于是它向你求助。

Input

输入分为三行:

第一行为两个数字$n(1 \le n \le 200000)$，$k(0 \le k \le n-1)$。

第二行为两个数字$l,r, 1 \le l \le r \le n$，代表Bbaka给出的限制区间，此处l,r的取值不固定。

第三行为n个数字，代表初始序列p，输入保证p为1到n的一个排列。

Output

输出一个整数，代表让初始序列p变为序列p'，$f(p')=k$，且每次操作都符合Bbaka的限制下所需的最小操作数。

若不可能构造出满足题意得序列p'，则输出-1。

Examples

standard input	standard output
4 0 1 2 2 1 3 4	1
4 0 1 2 1 4 2 3	2

Note

对于第一个样例，只需要交换$p[1]$与$p[2]$，显然1在区间$[1,1]$中，故这是合法的操作。最终$p'=\{1,2,3,4\}$，显然$f(p')=0$满足题意，且不存在花费操作数更少的构造。

对于第二个样例，只需要交换$p[2]$与$p[4]$，再交换一次$p[2]$与$p[3]$即可，显然两次交换中都有一个操作数2在区间$[1,2]$中，故这两次操作都是合法的。最终$p'=\{1,2,3,4\}$，显然$f(p')=0$满足题意，且不存在花费操作数更少的构造。