ソート (Sorting)

Aizhan は長さ N の数列 $S[0], S[1], \ldots, S[N-1]$ を持っている．この数列は 0 から $N-1$ までの相異なる数からなる．彼女は要素のペアをいくつか入れ替えることによって，この数列を昇順にソートしようとしている．彼女の友達の Ermek もこの数列の要素のペアをいくつか入れ替えようとしているが，Aizhan がソートする助けにはならないかもしれない．

Ermek と Aizhan はこの数列をいくつかのラウンドの間に並べ替えようとしている．各ラウンドでは，始めに Ermek が入れ替え操作を行い，次に Aizhan が入れ替えを行う．入れ替えを行う人は 2 つの添字を選び，それらに対応する要素を入れ替える．このとき，2 つの添字が異なる必要はないことに注意せよ．もし選んだ 2 つの添字が等しい場合は，要素は自分自身と入れ替えられるため，数列に変化はない．

Aizhan は Ermek が数列 S をソートすることには興味がないことを知っている．さらに彼女は，Ermek が選ぼうとしている添字を知っている．Ermek が M 回のラウンドを行おうとしていることも知っている．0 以上 $M-1$ 以下の各 i に対し，Ermek はラウンド i に $X[i]$ と $Y[i]$ の2 つの添字を選ぶ．

Aizhan は数列 S をソートしたい．各ラウンドに対し，もしラウンドの前に数列が昇順にソートされていたならば，彼女は作業を終了し，2 人はそれ以上の入れ替え操作を行わない．あなたは，元の数列 S と Ermek が選ぼうとしている添字が与えられる．あなたの課題は，入れ替え操作の列であって，Aizhan が行うことで数列 S をソートできるようなものを見つけることである．さらに，いくつかの小課題ではあなたは，このような操作の列のうち最も短いものを見つける必要がある．数列 S を M 回以下のラウンドでソートすることが可能であることを仮定してよい．

Ermek が入れ替え操作を行った直後に数列 S がソートされた場合，Aizhan は 2 つの同じ添字 (例えば 0 と 0 など) を選ぶことができることに注意せよ．この場合，ラウンド終了後にも S はソートされた状態になり，Aizhan は目的を達成できる．また，始めから数列 S がソートされていた場合は，ソートするのに必要なラウンド数は 0 となることに注意せよ．

例1 (Example 1)

以下の状況を考える:

- 始めの状態では，$S = 4, 3, 2, 1, 0$ である．
- Ermek は $M = 6$ 回の入れ替え操作をしようとしている．
- Ermek が選ぼうとしている添字を表す数列 X と Y は，$X = 0, 1, 2, 3, 0, 1$，$Y = 1, 2, 3, 4, 1, 2$ である．つまり，Ermek が選ぼうとしている添字のペアは，$(0, 1)$，$(1, 2)$，$(2, 3)$，$(3, 4)$，$(0, 1)$，$(1, 2)$ である．

この場合，Aizhan は 3 ラウンドで数列 S を $0, 1, 2, 3, 4$ にソートすることができる．彼女

は $(0, 4)$, $(1, 3)$, $(3, 4)$ と添字を選べばよい.

以下の表は Ermek と Aizhan がどのように数列を並び替えるかを表している.

ラウンド	操作者	添字のペア	数列
始め			$4, 3, 2, 1, 0$
0	Ermek	$(0, 1)$	$3, 4, 2, 1, 0$
0	Aizhan	$(0, 4)$	$0, 4, 2, 1, 3$
1	Ermek	$(1, 2)$	$0, 2, 4, 1, 3$
1	Aizhan	$(1, 3)$	$0, 1, 4, 2, 3$
2	Ermek	$(2, 3)$	$0, 1, 2, 4, 3$
2	Aizhan	$(3, 4)$	$0, 1, 2, 3, 4$

例2 (Example 2)

以下の状況を考える:

- 始めの状態では, $S = 3, 0, 4, 2, 1$ である.
- Ermek は $M = 5$ 回の入れ替えをしようとしている.
- Ermek が選ぼうとしている添字のペアは, $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, $(0, 4)$ である.

この場合, Aizhan は 3 ラウンドで数列 S をソートすることができる. 例えば彼女は $(1, 4)$, $(4, 2)$, $(2, 2)$ と添字を選べばよい.

以下の表は Ermek と Aizhan がどのように数列を並び替えるかを表している.

ラウンド	操作者	添字のペア	数列
始め			$3, 0, 4, 2, 1$
0	Ermek	$(1, 1)$	$3, 0, 4, 2, 1$
0	Aizhan	$(1, 4)$	$3, 1, 4, 2, 0$
1	Ermek	$(4, 0)$	$0, 1, 4, 2, 3$
1	Aizhan	$(4, 2)$	$0, 1, 3, 2, 4$
2	Ermek	$(2, 3)$	$0, 1, 2, 3, 4$
2	Aizhan	$(2, 2)$	$0, 1, 2, 3, 4$

課題 (Task)

あなたは数列 S, 数 M および添字を表す数列 X, Y を与えられる. Aizhan が数列 S をソートするために使えるような, 要素の入れ替え操作の列を求めよ. 小課題 5 と 6 においては, あなたが求める要素の入れ替え操作の列の長さは可能な限り短くなければならない.

あなたは関数 `findSwapPairs` を実装しなければならない:

- findSwapPairs(N, S, M, X, Y, P, Q) ：この関数は採点用プログラムによってちょうど 1 回呼び出される.
  - N: 数列 S の長さ.
  - S: 最初の数列 S を表す整数の配列.
  - M: Ermek が行おうとしている入れ替え操作の回数.
  - X, Y: それぞれ長さ M の配列である．各 $0 \leqq i \leqq M-1$ に対して，Ermek はラウンド i において数列の添字 $X[i]$, $Y[i]$ の要素を入れ替えようとしていることを表す.
  - P, Q: 整数の配列である．Aizhan が数列 S をソートするために使える入れ替え操作の列を返すために，この配列を用いよ．あなたのプログラムが見つけた入れ替え操作の列の長さを R とする．各 $0 \leqq i \leqq R-1$ に対して，Aizhan がラウンド i にて選ぶべき添字は $P[i], Q[i]$ に入っていなければならない．配列 P および Q は，M 要素分はすでに確保されていると仮定してよい.
  - この関数は，上に示した R を戻り値として返さなければならない.

小課題 (Subtasks)

小課題	点数	N	M	X, Y の追加の制約	R の条件
1	8	$1 \leqq N \leqq 5$	$M = N^2$	全ての i について $X[i] = Y[i] = 0$	$R \leqq M$
2	12	$1 \leqq N \leqq 100$	$M = 30N$	全ての i について $X[i] = Y[i] = 0$	$R \leqq M$
3	16	$1 \leqq N \leqq 100$	$M = 30N$	全ての i について $X[i] = 0, Y[i] = 1$	$R \leqq M$
4	18	$1 \leqq N \leqq 500$	$M = 30N$	なし	$R \leqq M$
5	20	$6 \leqq N \leqq 2{,}000$	$M = 3N$	なし	可能な限り最小
6	26	$6 \leqq N \leqq 200{,}000$	$M = 3N$	なし	可能な限り最小

M 回以下のラウンドでソートを行うような解が必ず存在することを仮定してよい.

採点用プログラムのサンプル (Sample grader)

採点用プログラムのサンプルは，次のフォーマットに従って sorting.in から入力を読み込む：

- 1 行目: N
- 2 行目: S[0] … S[N - 1]
- 3 行目: M

- 4 行目から `M` + 3 行目: `X[i] Y[i]`

採点用プログラムのサンプルは以下のように出力をする:

- 1 行目: `findSwapPairs` の戻り値 R
- 0 以上 $R-1$ 以下の i に対し，2+i 行目: `P[i] Q[i]`