Sorting

Aizhan tiene una secuencia de N enteros $S[0], S[1], \ldots, S[N-1]$. La secuencia está conformada por números distintos desde 0 a $N-1$. Ella está intentando ordenar la secuencia en orden ascendente intercambiando algunos pares de elementos. Su amigo Ermek también va a intercambiar algunos elementos — No necesariamente para ayudar a ordenar la secuencia.

Ermek y Aizhan van a modificar la secuencia en una serie de rondas. En cada ronda, primero Ermek hace un intercambio y luego Aizhan realiza otro intercambio. De forma más precisa, la persona que hace el intercambio elije dos índices válidos dentro de la secuencia y realiza el intercambio de los elementos ubicados en dichos índices. Note que no necesariamente los índices deben ser diferentes. En el caso de que ambos sean iguales, la persona realiza el intercambio de un elemento consigo mismo, lo cual, no altera la secuencia.

Aizhan sabe que a Ermek no le interesa mucho ordenar la secuencia S. Ella también sabe exactamente que índices planea utilizar Ermek para hacer los intercambios. Ermek tiene pensado participar en M rondas de intercambio. Las rondas se encuentran numeradas de 0 a $M-1$. Por cada i entre 0 y $M-1$ inclusive, Ermek va a elegir los índices $X[i]$ y $Y[i]$ en la ronda i.

Aizhan desea ordenar la secuencia S. Antes de cada ronda, si Aizhan ve que la secuencia está ordenada ascendentemente, ella finaliza el proceso completo. Dada la secuencia original S y los índices que Ermek va a elegir, su tarea es encontrar una secuencia de intercambios que pueda utilizar Aizhan para ordenar la secuencia S. Además, en algunas subtareas se requiere que encuentre la secuencia de intercambios más corta posible. Si hay más de una solución óptima, usted puede retornar cualquiera de ellas. Además usted puede asumir que siempre es posible ordenar la secuencia S en M o menos rondas.

Observa que si Aizhan ve que la secuencia S está ordenada luego del intercambio realizado por Ermek, ella puede elegir intercambiar dos índices iguales (por ejemplo 0 y 0). Como resultado, la secuencia S estaría también ordenada luego de la ronda completa, y Aizhan lograría su objetivo. También nota que si la secuencia inicial S ya se encuentra ordenada, el mínimo número de rondas requeridas para ordenar dicha secuencia es 0.

Example 1

Suponga que:

- La secuencia inicial es $S = 4, 3, 2, 1, 0$.
- Ermek está dispuesto a hacer $M = 6$ intercambios.
- Los arreglos X y Y que describen los intercambios que va a realizar Ermek son $X = 0, 1, 2, 3, 0, 1$ y $Y = 1, 2, 3, 4, 1, 2$. En otras palabras, los pares de índices que Ermek planea utilizar en sus intercambios son $(0, 1)$, $(1, 2)$, $(2, 3)$, $(3, 4)$, $(0, 1)$, y $(1, 2)$.

Con estos datos Aizhan puede ordenar la secuencia S y convertirla en $0, 1, 2, 3, 4$ en tres rondas.

Ella puede hacerlo también si elige los índices $(0, 4)$, $(1, 3)$, y luego $(3, 4)$.

La siguiente tabla muestra como Ermek y Aizhan modifican la secuencia.

Ronda	Jugador	Par de índices intercambiados	Secuencia
Inicio			$4, 3, 2, 1, 0$
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	$(3, 4)$	$0, 1, 2, 3, 4$

Example 2

Suponga que:

- La secuencia inicial es $S = 3, 0, 4, 2, 1$.
- Ermek está dispuesto a hacer $M = 5$ intercambios.
- Los pares de índices que Ermek planea utilizar en sus intercambios son $(1, 1)$, $(4, 0)$, $(2, 3)$, $(1, 4)$, y $(0, 4)$.

Con estos datos Aizhan puede ordenar la secuencia S en tres rondas, por ejemplo eligiendo los pares de índices $(1, 4)$, $(4, 2)$, y luego $(2, 2)$. La siguiente tabla muestra como Ermek y Aizhan modifican la secuencia.

Ronda	Jugador	Par de índices intercambiados	Secuencia
Inicio			$3, 0, 4, 2, 1$
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	$(2, 2)$	$0, 1, 2, 3, 4$

Task

A usted le será dada la secuencia S, el número M, y las secuencias de índices X y Y. Calcule una secuencia de intercambios, con los cuales Aizhan pueda ordenar la secuencia S. En las subtareas 5 y 6 la secuencia de cambios que usted debe encontrar debe ser la menor posible.

Usted debe implementar la función `findSwapPairs`:

- `findSwapPairs(N, S, M, X, Y, P, Q)` — Esta función será invocada por el grader exactamente una vez.

- `N`: la longitud de la secuencia S.
- `S`: un arreglo de enteros con la secuencia inicial S.
- `M`: el número de intercambios que Ermek está dispuesto a hacer.
- `X, Y`: arreglos de enteros de longitud M. Para $0 \leq i \leq M-1$, en la ronda i Ermek planea intercambiar los números en los índices $X[i]$ y $Y[i]$.
- `P, Q`: arreglos de enteros. Utilice estos arreglos para retornar una de las posibles secuencias de intercambios que Aizhan debe realizar para lograr ordenar la secuencia S. Sea R la longitud de la secuencia de pasos que su programa ha encontrado. Para cada i entre 0 y $R-1$ inclusive, los índices que Aizhan debe elegir en la ronda i deben ser almacenados en los arreglos $P[i]$ y $Q[i]$. Usted puede asumir que los arreglos `P` y `Q` ya poseen el espacio suficiente para almacenar M elementos cada uno.
  - Esta función debe retornar el valor de R (definido anteriormente).

Subtasks

subtask	puntos	N	M	restricciones adicionales para X, Y	requisitos sobre R
1	8	$1 \leq N \leq 5$	$M = N^2$	$X[i] = Y[i] = 0$ Para todo i	$R \leq M$
2	12	$1 \leq N \leq 100$	$M = 30N$	$X[i] = Y[i] = 0$ Para todo i	$R \leq M$
3	16	$1 \leq N \leq 100$	$M = 30N$	$X[i] = 0, Y[i] = 1$ Para todo i	$R \leq M$
4	18	$1 \leq N \leq 500$	$M = 30N$	ninguno	$R \leq M$
5	20	$6 \leq N \leq 2,000$	$M = 3N$	ninguno	mínimo posible
6	26	$6 \leq N \leq 200,000$	$M = 3N$	ninguno	mínimo posible

Usted puede asumir que existe una solución que requiere M o menos rondas.
You may assume that there exists a solution that requires M or fewer rounds.

Implementation details

Usted debe enviar exactamente un archivo, llamado `sorting.c`, `sorting.cpp`, `sorting.pas`, or `sorting.java`.

Este archivo debe implementar la función descrita anteriormente utilizando alguna de las siguientes firmas.

C/C++ program (incluya `sorting.h` al comienzo de su archivo fuente)

```
int findSwapPairs(int N, int S[], int M, int X[], int Y[], int P[], int Q
```

Pascal programs (implement the described method in the unit `sorting`)

```
function findSwapPairs(N: longint; var S: array of longint, var M: longir
```

Java programs (implement the described method in the public class `sorting`)

```
public int findSwapPairs(int N, int S[], int M, int X[], int Y[], int P[]
```

Sample grader

El grader de ejemplo lee la entrada del archivo `sorting.in` de la siguiente manera:

- línea 1: `N`
- línea 2: `S[0]` … `S[N - 1]`
- línea 3: `M`
- líneas 4, …, `M` + 3: `X[i] Y[i]`

El grader de ejemplo imprime la siguiente información:

- línea 1: el valor de retorno R de la función `findSwapPairs`
- líneas 2+i, Para $0 \leq i < R$: `P[i] Q[i]`