



Depot Rearrangement (reorganizaci3n de un almac3n)

Input file: depot.in**100 points****Output file:** depot.out**Time limit: 2 sec****Source Code:** depot.pas / .c / .cpp**Memory limit: 64 MB**

Una empresa opera en N tiendas, vendiendo M diferentes productos en cada tienda. La empresa tiene un gran almac3n donde empaqueta los productos antes de distribuirlos a las tiendas. Cada tienda recibe el mismo n3mero de unidades de cada producto. La empresa empaqueta un cierto n3mero de unidades de un determinado producto en un contenedor, y etiqueta al contenedor con el identificador del producto. Los productos est3n identificados con los n3meros de 1 a M . De esta forma, al terminar de empaquetar hay $N \cdot M$ contenedores en el almac3n, y exactamente N contenedores est3n etiquetados con un determinado identificador de producto. Ya que el almac3n es un edificio estrecho, los contenedores se distribuyen en una sola fila. Para acelerar la distribuci3n, el encargado del almac3n quiere reorganizar a los contenedores. Como los productos se distribuyen a las tiendas enviando un solo camión a cada tienda, y cada camión transporta un contenedor de cada producto, una posible reorganizaci3n es la siguiente. Los primeros M contenedores de la fila deben etiquetarse con diferentes identificadores de productos, los siguientes M contenedores de la fila deben etiquetarse con diferentes identificadores de productos, y as3 sucesivamente. Desgraciadamente solo hay un lugar libre al final de la fila para poner a un contenedor. Entonces la reorganizaci3n debe realizarse por sucesivos movimientos en los que se coge un contenedor y se mueve a un lugar libre. Despu3s de la reorganizaci3n el lugar libre debe estar al final de la fila.

El objetivo es obtener la reorganizaci3n requerida con el m3nimo n3mero de movimientos.

Task

Escribe un programa que calcule la reorganizaci3n con el m3nimo n3mero de movimientos.

Input

La primera l3nea del fichero de texto depot.in contiene dos enteros, N y M . N ($1 \leq N \leq 400$) es el n3mero de tiendas y M ($1 \leq M \leq 400$) es el n3mero de productos. La segunda l3nea contiene $N \cdot M$ enteros, los identificadores de los contenedores en el orden inicial. Cada identificador de un producto x ($1 \leq x \leq M$) aparece exactamente N veces en la l3nea.

Output

La primera l3nea del fichero de texto depot.out contiene un entero S , el m3nimo n3mero de movimientos necesarios para obtener el orden requerido en la fila de contenedores (Subtask A). Las siguientes S l3neas describen la reorganizaci3n (Subtask B). Cada l3nea contiene un par de enteros x y y . El par x y describe un movimiento: el contenedor de la posici3n x se mueve a la posici3n y . Las posiciones se identifican con n3meros de 1 hasta $N \cdot M + 1$; inicialmente la posici3n $N \cdot M + 1$ est3 libre (en ella no hay ning3n contenedor). Un movimiento desde x hasta y es legal si la posici3n y est3 libre antes del movimiento. Despu3s de un movimiento de x a y la posici3n x est3 libre. Es suficiente mostrar la primera l3nea si solo solucionas la Subtask A.

Si hay m3ltiples posibilidades, tu programa solo debe mostrar a una de ellas.



CENTRAL-EUROPEAN OLYMPIAD IN INFORMATICS

Sárospatak, Hungary
28 July - 4 August 2005

Page 2 of 2

Español

Day 1: depot

Example

depot.in	depot.out
5 6	8
4 1 3 1 6 5 2 3 2 3 5 6 2 1 4 5 6 4 1 3 2 4 5 5 1 2 3 4 6 6	9 31
	18 9
	10 18
	4 10
	31 4
	30 31
	24 30
	31 24