



Mobile Service (servicio móvil)

Input file: service.in

100 points

Output file: service.out

Time limit: 3 sec

Source Code: service.pas/.c/.cpp

Memory limit: 64 MB

Una empresa presta servicios a sus clientes que están situados en diferentes ciudades. La compañía tiene tres empleados de servicio móvil. Si se produce una demanda de servicio en una posición, un empleado del servicio tiene que moverse desde su posición actual hasta la posición donde se ha producido la demanda (si allí no se encuentra ningún empleado) y así satisfacer a la demanda. En cualquier momento, solo se puede mover un empleado. Ellos solo se pueden mover si hay una demanda y no pueden estar en la misma posición. El movimiento de un empleado desde la posición p hasta la posición q tiene un coste $C(p,q)$. La función del coste no es necesariamente simétrica, pero si no hay movimiento el coste es 0 , por ejemplo $C(p,p)=0$. La empresa debe satisfacer las demandas recibidas en un estricto "la primera demanda en llegar es la primera en servirse". El objetivo es minimizar el coste total del servicio de una secuencia de demandas.

Task

Debes escribir un programa que decida que empleado del servicio debe moverse para cada demanda, de manera que el coste total del servicio a la secuencia de demandas sea mínima.

Input

La primera línea del fichero de texto `service.in` contiene dos enteros, L y N . L ($3 \leq L \leq 200$) es el número de posiciones y N ($1 \leq N \leq 1000$) es el número de demandas. Las posiciones se identifican mediante números, de 1 a L . Cada una de las siguientes L líneas contiene L enteros no negativos. El j -ésimo número de la línea $i+1$ tiene de coste $C(i,j)$, y es menor de 2000 . La última línea contiene N enteros, la lista de las demandas. Una demanda se identifica por el identificador de la posición donde ocurre la demanda. Inicialmente, los tres empleados están en la posición 1 , 2 y 3 , respectivamente.

Output

La primera línea del fichero de texto `service.out` contiene un entero, M , el coste mínimo de dar servicio a una secuencia de demandas (subtask A). La segunda línea contiene exactamente N enteros. El i -ésimo número es el identificador del empleado (1 , 2 o 3) que servirá a la i -ésima demanda (subtask B). Si hay diversas posibilidades, tu programa solo debe mostrar a una de ellas. Puedes omitir la segunda línea si solo solucionas la Subtask A.



CENTRAL-EUROPEAN OLYMPIAD IN INFORMATICS

Sárospatak, Hungary
28 July - 4 August 2005

Page 2 of 2

Español

Day 1: service

Example

service.in	service.out
5 9	5
0 1 1 1 1	1 2 1 2 2 1 3 1 3
1 0 2 3 2	
1 1 0 4 1	
2 1 5 0 1	
4 2 3 4 0	
4 2 4 1 5 4 3 2 1	