

Problem E: If only I could think Linearly...

Any *true* algorithm junkie knows that a linear system with multiple inputs (vector \mathbf{x}) and multiple outputs (vector \mathbf{y}) can be characterized by a matrix \mathbf{M} . Each column of \mathbf{M} contains the values of the system outputs when the input with the same number as that column is unity, and all other inputs are zero. Because the system is linear, the outputs for an arbitrary configuration of inputs can be obtained from a linear combination of the columns of \mathbf{M} , using the equation:

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

Your challenge is to determine the inputs (\mathbf{x}) that produce a given output (\mathbf{y}), given the matrix (\mathbf{M}). Your solution must read the matrix \mathbf{M} from a text file that contains the number of rows (m , an integer) on the first line, the number of columns (n , an integer) on the second line, and one row of the matrix for each following line (floating point numbers separated by whitespace). It must then read the output vector which contains one output value per line (m floating point numbers). There is a single test case in the input file.

In other words, you are being given the matrix \mathbf{M} of dimensions $m \times n$ and the column vector \mathbf{y} of dimension m . Your assignment is to discover the column vector \mathbf{x} of dimension n (with exactly three non-zero entries) that makes the following equation true (shown with $m = 3$ and $n = 4$):

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

Your solution must handle cases where the number of inputs (n) is larger than the number of outputs (m). In other words, the matrix \mathbf{M} is underdetermined and the system has a many-to-one mapping (more than one input, \mathbf{x} , can produce a given output, \mathbf{y}). In this case, however, you are guaranteed that \mathbf{x} has exactly 3 non-zero entries. Your solution should be printed to standard out and should give the indices of the non-zero entries in \mathbf{x} as integer values 1 through n , inclusive, along with the value of those three inputs. An example problem is:

Sample Input

5

8

0.484734 -0.147660 -0.224671 -0.060712 -0.324758 0.209435 -0.461233 -0.402469
-0.215098 0.008685 -0.254425 0.423113 -0.033949 0.003395 0.126518 -0.099603
0.048778 0.156349 0.228357 0.068926 0.418765 -0.480551 -0.026597 -0.202028
-0.317657 -0.181216 -0.043709 -0.207515 0.396223 0.028521 -0.091552 0.419526
-0.000758 0.163841 0.294449 -0.223812 -0.456281 0.239201 0.452864 0.284574
0.034654
-2.504212
-4.200988
3.153023
3.517523

Sample Output

input 1 = -4.0365696

input 4 = -8.028227

input 6 = 7.180792

Matrix **M**, when multiplied by your solution for **x**, should be able to reproduce each entry in vector **y** to within 0.01%. The three indices that you output need not appear in increasing order, but they must match the correct solution exactly. The value you print for each index must be correct to 3 significant digits, but you are free to output extra digits