

Treasure Hunt

34	21	32	41	25
14	42	43	14	31
54	45	52	42	23
33	15	51	31	35
21	52	33	13	23

You are going to write a program to explore the above table for a treasure. The values in the table are clues. Each cell contains a number between 11 and 55, where the ten's digit represents the row number and the unit's digit represents the column number of the cell containing the next clue. Starting with the upper left corner (at 1,1), use the clues to guide your search through the table - (the first three clues are 11, 55, 15). The treasure is a cell whose value is the same as its coordinates. Your program must first read in the treasure map data into a 5 by 5 array.

Preferred Languages

Python, Groovy, Scala, Ruby, (Vanilla) Javascript

Input Format

Input contains five lines each containing five space separated integers.

Output Format

If the treasure is found, your program should output the index (row, column) of cells it visits during its search for treasure (separated by a single space). Cells must be separated by a newline “\n”.

If there is no treasure, print “NO TREASURE”

Implementation

Write two different implementations. The first should use a functional programming approach (closures, native datastructures). The second implementation should be implemented in an object-oriented way (object models, simple oo patterns). One of the implementations should be coded with recursion, the other without recursion.

For non javascript: Read input from STDIN. Print output to STDOUT. Do not use external libraries.

For Javascript: Input is a list of tuples (array of arrays), while the output can be logged to the console.

Sample Input

```
55 14 25 52 21
44 31 11 53 43
24 13 45 12 34
42 22 43 32 41
51 23 33 54 15
```

Sample Output

```
1 1
5 5
```

15
21
44
32
13
25
43