

# Practica 1

El objetivo de la práctica es validar los conceptos vistos en clase a través de tres ejercicios sencillos que deben ser implementados en Python. El proceso para la resolución de estos problemas debe comenzar desde la abstracción del mismo y debe pasar por el diseño de algoritmos que al menos resuelvan el problema.

## Problema 1

Dada una matriz, decimos que está ordenada por filas si todos los números de cada fila están en orden creciente. Del mismo modo, está ordenada por columnas si todos los números de cada columna están ordenados crecientemente. En este problema es necesario que digas si un número  $x$  se encuentra en una matriz ordenada por filas y columnas, y si está, en qué posición.

**Entrada y salida:** La entrada empieza con el número de filas  $n$  y columnas  $m$  y el número a buscar  $x$ . Tras esto viene  $n$  líneas con  $n$  enteros positivos cada una. Se garantiza que la matriz es cuadrada y por tanto  $n=m$ .

A continuación empieza el diálogo con el corrector. Tu programa puede escribir dos tipos de instrucciones.

- “? a b”: Pregunta por el elemento en la fila  $a$  y columna  $b$ . Recibirás por la entrada el número que ocupa esa posición.
- “= a b”: Escribe la posición en la que se encuentra  $x$ . Si no está en la matriz, utiliza  $a = b = -1$ .

Debe tener en cuenta que tras enviar una instrucción de tipo “=”, el corrector dejará de corregir. El programa no puede escribir más de 250 instrucciones. Cada instrucción en una línea con DOS saltos de línea al final.

### Archivos necesarios

Archivo de entrada y corrector. El corrector los subiré al github que hemos usado en clase.

### Puntuación

- Caso 1: Resolver casos con  $n = 5$ .
- Caso 2: Resolver casos con  $n = 20$ .

- Caso 3: Resolver casos con  $n = 50$ .
- Caso 4: Resolver casos con  $n = 100$ .

Cada caso vale 25 puntos, para contabilizar 100 puntos.

## Problema 2

Hacer un programa que determine, para cada palabra dada, si está correctamente parentizada o no. Aquí, una palabra es una secuencia no vacía de ‘(’, ‘)’, ‘[’ y ‘]’ sin espacios intermedios.

Por ejemplo éstas son algunas palabras correctamente parentizadas:

( ) [] O[] [O] ((O)) O[]([]) ([[]O](O))

y éstas son algunas palabras incorrectamente parentizadas:

) [] ([[] DO] [(O)) O[][]() ([[]O](O

**Entrada:** La entrada consiste en una serie de palabras de entre 1 y 1000 caracteres, una por línea.

**Salida:** Para cada palabra dada, hay que escribir “correcta” o “incorrecta” según convenga.

**Puntuación:**

- Caso 1: Resolver con palabras que solo contienen ‘(’ y ‘)’. **30 puntos.**
- Caso 2: Resolver palabras con los cuatro caracteres ‘(’, ‘)’, ‘[’ y ‘]’. **70 puntos.**

## Problema 3

Haga un programa que encuentre el mínimo número  $x_i$  que aparece por lo menos tantas veces como cualquier otro número en la sucesión  $x_1, x_2, \dots, x_n$ .

**Entrada:** Un archivo con la sucesión de números  $x_1, x_2, \dots, x_n$  uno en cada línea. Es decir el archivo contendrá un total de  $n$  líneas. Suponga que el archivo de entrada es tal que  $2 \leq n \leq 1000$  y que  $1 \leq x_i \leq 10000$