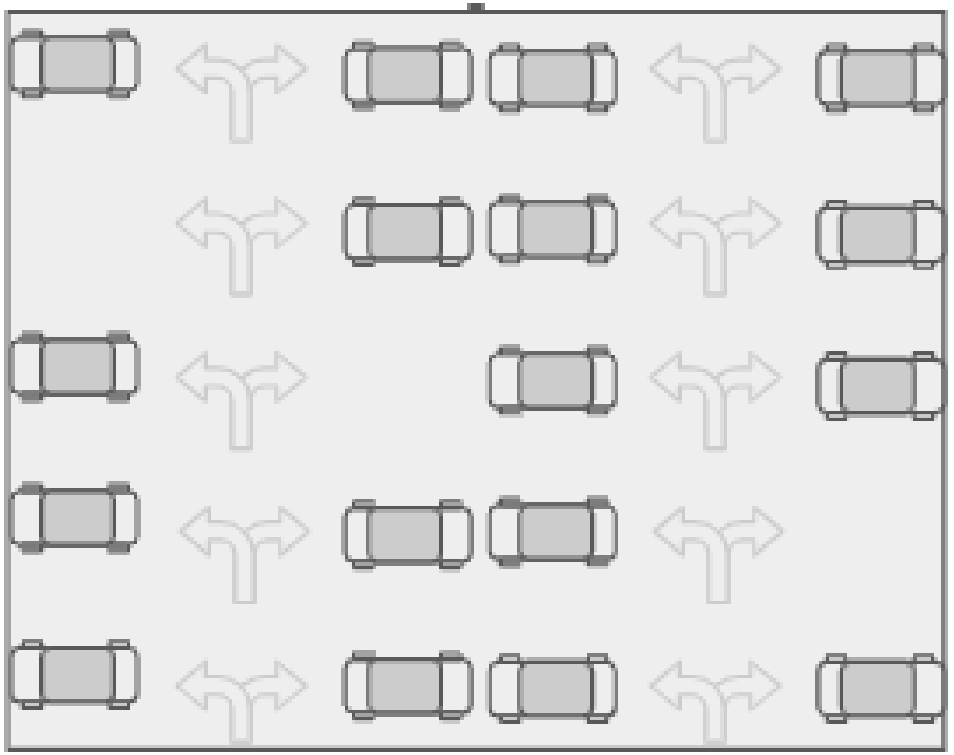
Programación I
Ejercicios de manejo avanzado de datos

Pablo Garaizar
Borja Sanz Urquijo
M.ª Luz Guenaga
Jenny Fajardo

Facultad de Ingeniería

Parking

- Crear una lista multidimensional para almacenar la información de ocupación de un parking:



Filas

- Programa una función que reciba un entero dimensión y devuelva una lista bidimensional de esa dimensión que tenga todo 1 en la primera fila, todo 2 en la segunda, etc:

[[1, 1, 1],
  [2, 2, 2],
  [3, 3, 3]]

Columnas

- Programa una función que reciba un entero dimensión y devuelva una lista bidimensional de esa dimensión que tenga todo 1 en la primera columna, todo 2 en la segunda, etc:

[[1, 2, 3],
  [1, 2, 3],
  [1, 2, 3]]

Matriz identidad

- Programa una función que reciba un entero dimensión y devuelva una lista bidimensional con la matriz identidad de esa dimensión:

$$I_1 = (1),\ I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},\ I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},\ \cdots,\ I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

Matrices

- Programa una función que reciba dos matrices de las mismas dimensiones y devuelva la matriz suma.
- Programa una función que reciba dos matrices y devuelva la matriz producto.

Limpia matrices

- Programa una función que reciba una matriz y devuelva la misma matriz eliminando las filas y/o columnas que tengan solo ceros:

$$\begin{pmatrix} 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 3 & 3 \\ 3 & 2 & 6 \\ 3 & 3 & 1 \\ 5 & 5 & 1 \end{pmatrix}$$

¿Es cuadrado mágico?

- Programa una función reciba una lista bidimensional y devuelva True si es un cuadrado mágico (todas sus filas y columnas suman lo mismo).

4	9	2
3	5	7
8	1	6

Determinantes

- Programa una función que reciba una matriz bidimensional de rango 3 y devuelva su determinante:

$$det\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = (a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32}) - (a_{13}a_{22}a_{31} + a_{12}a_{21}a_{33} + a_{11}a_{23}a_{32})$$

- Programa una función que reciba cualquier matriz cuadrada y devuelva su determinante.

Letras o no

- Programa una función que reciba una cadena de caracteres y devuelva una lista indicando si hay letras (a-z, A-Z) o no:

  letrasONo(‘abcd 12 ABC’)

  [1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1]

Tipos de caracteres

- Programa una función que reciba una cadena de caracteres y devuelva una lista indicando si son caracteres especiales (0), letras (1) o números (2):

  caracteres(‘abcd 12 ABC ?!#’)

  [1, 1, 1, 1, 0, 2, 2, 0, 1, 1, 1, 0, 0, 0, 0]

Polinomios

- Crea una lista de polinomios en la que cada polinomio será una lista con los coeficientes en orden.
  - Por ejemplo:

    [2, 3, -4, 0, 1] representa a $2x^4 + 3x^3 - 4x^2 + 1$

- Programa una función que recibe todos los polinomios y devuelve una lista con la suma de todos los polinomios.

Facturas

- Crea un diccionario donde la clave sea el código de factura y el valor la cantidad pendiente de cobro.
- Programa una función que reciba ese diccionario y devuelva el total pendiente de cobro.
- Programa una función que reciba ese diccionario y devuelva la factura con mayor importe.

Morse

- Crea un diccionario donde las claves sean letras y números y los valores sean su equivalente en código Morse.
- Crea una función que reciba un texto y el diccionario anteriormente creado y devuelva el mismo texto en código Morse.

Traducciones

- Crea un diccionario donde las claves sean palabras en inglés y los valores sus traducciones a castellano.
- Crea una función que reciba un texto en inglés y el diccionario anteriormente creado y devuelva el mismo texto sustituyendo cada palabra encontrada en el diccionario por su correspondiente en castellano.

Cuenta caracteres

- Programa una función que reciba una cadena de caracteres y devuelva un diccionario con el total de veces que aparece cada carácter:

  cuentaChars(‘abcd 11 abbb ?’)

  { ‘a’: 2, ‘b’: 4, ‘c’: 1, ‘d’: 1, ‘ ‘: 3, ‘1’: 2, ‘?’: 1 }

Cuenta palabras

- Programa una función que reciba una cadena de caracteres y devuelva un diccionario con el total de veces que aparece cada palabra:

  cuentaPalabras(‘hola hola que que hola adiós’)

  { ‘hola’: 3, ‘que’: 2, ‘adiós’: 1 }

Personajes

- Crea un diccionario para almacenar la posición en pantalla de cada personaje en un videojuego. La clave será el nombre del personaje y el valor será un diccionario con un valor para la clave x y otro para la clave y.
- Programa una función que reciba el diccionario de personajes y una posición (x, y) y devuelva el nombre del personaje que esté más cerca a esa posición.
- Programa una función que reciba el diccionario de personajes y devuelva un diccionario con los dos personajes que se encuentran más cerca entre sí.

Agenda

- Crea un diccionario para almacenar los datos de una agenda de contactos (nombre, apellidos, teléfono, email).
- Programa funciones para gestionar las altas, bajas y modificaciones de la agenda.
- Programa funciones para buscar contactos por nombre y por teléfono en la agenda.