

Grado en Diseño y Desarrollo de Videojuegos
Algoritmos para Juegos
Práctica 1: Comenzando a programar con Python

1.- Supón que, para un juego, tienes que calcular el desglose mínimo en billetes y monedas de una cantidad exacta de euros. Hay billetes de 500, 200, 100, 50, 20, 10 y 5 € y monedas de 2 y 1 €. Por ejemplo, si deseamos conocer el desglose de 434 €, el programa mostrará por pantalla el siguiente resultado:

```
2 billetes de 200 euros
1 billete de 20 euros
1 billete de 10 euros
2 monedas de 2 euros
```

2.- Recordáis el juego de la patata caliente en un programa de TV que se llamaba Grand Prix? Vamos a simularlo con Python. El programa generará un número aleatorio entre 1 y N (puede ser de utilidad la función random del módulo random). El juego consiste en que el usuario debe adivinar el número, introduciendo por teclado una propuesta para el número y esperando la respuesta del programa, que le dirá si el número es mayor o menor que el que se ha de adivinar. El programa debe terminar cuando el usuario acierta el número.

3.- Escribir un programa que lea un número positivo y escriba el triángulo de Floyd hasta el número leído. Por ejemplo, si se introduce el 8, el resultado en pantalla debe ser:

```
1
2   3
4   5   6
7   8
```

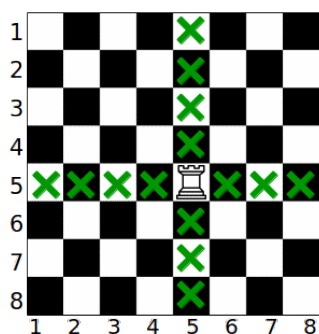
4.- El juego del ahorcado consiste en que un jugador (A) debe adivinar la palabra pensada por el jugador (B). Para ello debemos utilizar dos cadenas de caracteres: en la primera se guardará la palabra que ha pensado el jugador (B), y la segunda se actualizará en cada turno para contemplar los aciertos del jugador (A). Esta segunda cadena debe inicializarse a una cadena de caracteres de la misma longitud que la primera pero rellena con caracteres como: '_' ó '*'. El jugador (A) tiene una cantidad máxima de fallos permitidos (elegido por el programador). Se considera fallo preguntar por un carácter que no se encuentra en la palabra pensada por el jugador (B). El jugador (A) gana la partida cuando acierta la palabra sin consumir el total de fallos permitidos. El jugador (B) gana cuando el jugador (A) consume todos los fallos permitidos.

5.- Crear de manera automática un tablero de damas donde las posiciones ocupadas inicialmente en una partida contengan el valor 1 para las fichas de un jugador y 2 para el otro. Las posiciones desocupadas contendrán el valor 0.

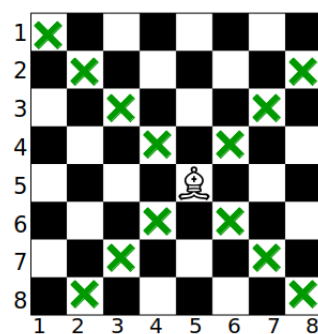
1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	2	0	2	0	2	0	2
2	0	2	0	2	0	2	0
0	2	0	2	0	2	0	2

Fig 1: Configuración inicial del tablero para el juego de las damas.

6.- En ajedrez, una torre se puede mover desde la posición que ocupa, a cualquier casilla de su fila o columna, mientras que el alfil se mueve sobre el tablero en una línea recta diagonal, tal y como se muestra en la Fig. 2. Se pide realizar un programa que pida al usuario una ficha (torre o alfil) y el índice de fila y de columna que ocupa, e imprima por pantalla un tablero en la que se marquen las casillas a las que puede moverse la ficha.



(a)



(b)

Fig 2.: Un ejemplo de movimiento para (a) la torre y (b) el alfil en un tablero de ajedrez.

7.- Implementa un programa que pida al usuario un número N entre 1 y 10 y que genere e imprima por pantalla una matriz de tamaño NxN en el que los bordes de la matriz deben valer 0 y el resto, deben guardar la distancia al borde más cercano. Ejemplo: Para un valor de la entrada de 7, la matriz generada debe ser:

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

8.- Implementa un programa que pida al usuario un número N entre 1 y 10 y que genere e imprima por pantalla una matriz de tamaño NxN en el que cada elemento debe valer los números naturales consecutivos, comenzando desde la última posición hasta el inicio, en el orden de las columnas. Ejemplo: Para un valor de la entrada de 4, la matriz generada debe ser:

16	12	8	4
15	11	7	3
14	10	6	2
13	9	5	1

9.- Implementa un programa que pida al usuario un número N entre 1 y 10 y que genere e imprima por pantalla una matriz de tamaño NxN en el que cada elemento debe valer los números naturales consecutivos, comenzando desde la última posición hasta el inicio. Ejemplo: Para un valor de la entrada de 4, la matriz generada debe ser:

16	15	14	13
12	11	10	9
8	7	6	5
4	3	2	1

10.- Implementa un programa que pida al usuario un número entre 1 y 10 y que genere e imprima por pantalla una matriz de ese tamaño que almacene un número entero consecutivo por cada una de las diagonales que se pueden trazar partiendo desde el origen. El primero de los números debe ser 0. Ejemplo: Para el número 5, la matriz generada debe ser:

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
3	4	5	6	7
4	5	6	7	8

11.- Implementa un programa que pida al usuario un número entre 1 y 15 y que genere e imprima por pantalla una matriz 5x5 que almacene el triángulo de Floyd en sentido inverso de las filas para ese número, poniendo ceros en el resto de las posiciones. Ejemplo: Para el número 13, la matriz generada debe ser:

11	12	13	0	0
7	8	9	10	0
4	5	6	0	0
2	3	0	0	0
1	0	0	0	0

12.- Realice un programa que pida al usuario una frase y que utilice un diccionario para contar el número de veces que aparece cada letra. Si una letra no aparece, no será necesario decir que aparece 0 veces.

13.- Implemente un programa en Python que utilice un diccionario para traducir a valores numéricos las palabras que hacen referencia a números de una frase introducida por teclado.

14.- Implemente un programa en Python que reciba una frase y que utilice un diccionario para contar el número de ocurrencias de cada palabra en la frase.

15.- Juego de las 4 en línea: El juego de las cuatro en línea se juega en una estructura vertical de 7 filas y 8 columnas en la que dos jugadores van colocando fichas en cada columna. Las fichas caen, por efecto de la gravedad, hasta que llegan al fondo de la estructura o hasta que descansan sobre otra ficha colocada antes. Se pide hacer un simulador de este juego en Python que permita jugar a dos jugadores y que compruebe tras cada jugada, si alguno ha hecho cuatro en raya. Para ello, es conveniente dividir el programa en diferentes subprogramas que se vayan encargando de cada una de las tareas necesarias.