3. Control de flujo

Ejercicio12

Diseñar una función para que dado un vector x, genere otro vector con el orden de los elementos invertido.

a) Diseñar la función utilizando bucles for.

b) Diseñar la función utilizando bucles $while$.

Ejercicio13

Completar los siguientes apartados

1. Diseña un función que dado un vector x, genere una lista con la resta entre elementos consecutivos.
2. Diseña un función que dado un vector x, genere una lista con las sucesivas sumas acumuladas de los elementos del vector.

Ejercicio14

Diseña una función que dado un numero a y un vector x, extraiga los a primeros elementos del vector x, y los añada al final en el mismo orden que fueron extraídos.

Ejercicio15

Completar los siguientes apartados:

1. Diseñar una función que calcule la $media$ de un vector utilizando bucles $while$.
2. Diseñar una función que calcula la desviación standard de un vector utilizando bucles for.
3. Crear un script para combinar el uso de ambas funciones de manera que dado un vector se obtenga su media y desviación estandard.

La formula de la media es:

$$\bar{v} = \frac{\sum_{i=1}^{n} v_i}{n}$$

La formula de la desviación es:

$$stdv = \sqrt{\frac{\sum_{i=1}^{n} (v_i - \bar{v})^2}{n}}$$

Ejercicio16

El dilema del prisionero es un caso clásico de teoría de juegos. El juego versa sobre la cooperación, dos prisioneros A y B son interrogados por la policía de manera independiente, de tal modo que la pena de cada uno de los prisioneros depende de la confesión del otro. Las reglas se detallan en la siguiente tabla.

	A confiesa	A miente
B confiesa	A y B son condenados a 1 año.	A sale libre y B es condenado a 3 años.
B miente	A es condenado a 3 años y B sale libre.	Ambos son condenados a 2 años.

Diseñar una función en matlab, utilizando la función *switch* que implemente las reglas del dilema del prisionero. La función recibirá dos entradas A y B que representan las decisiones de los prisioneros, siendo $1 = confesión$ y $0 = mentira$. La función debe devolver las penas asignadas a cada prisionero.

Ejercicio17

Diseñar una función que devuelva la mayor de la suma de las columnas de una matriz A.

Ejercicio18

Implementar una función que tome como entrada una matriz A y que devuelva la suma de todos los elementos de las 2 diagonales de dicha matriz. No utilizar la función *diag*.

Ejercicio19

Implementar una función que dadas dos matrices A y B, implemente la multiplicación de matrices C=A·B mediante bucles.

Ejercicio20

Crea la función dec2bin_entero que ante la entrada de un numero entero devuelva un vector con el contenido de los números en binario. NOTA: no utilizar la función bin2dec.

Ejercicio21

Crea la función bin2dec_entero que realice el funcionamiento inverso, dado un vector en binario devuelva su valor en decimal.

Ejercicio22

Completar los siguientes apartados:

a) Diseñar una función que indique la posición de la primera aparición de un elemento e en un vector v. La función devolverá -1 si el elemento no se encuentra en el vector.

b) Sabiendo que la asignación $v(i) = []$ elimina el elemento que se encuentra en la posición i del vector. Escribir un script que utilizando la función previamente diseñada encuentre todas las apariciones de un elemento e en un vector v.

Ejercicio23

Diseñar una función que ordene los elementos de un vector, de menor a mayor, mediante un procedimiento de intercambio. La ordenación por intercambio consiste en dados dos elementos de un vector, intercambiarlos si la posición que ocupa el elemento menor es mayor que la ocupada por el elemento mayor.

Ejercicio24

Diseña una función que extraiga el elemento mas pequeño de un vector x.

Ejemplo:

$>> [elem, vec] = minimo([1, 2, 3, 0])$
$>> elem = 0$
$>> vec = [1, 2, 3]$

Utiliza dicha función para escribir un script que ordene un vector.