

Fundamentos de la Programación (Física - 2021/2022)

Trabajo de evaluación continua puntuable para la calificación final de la asignatura, con una ponderación del 5 %

Fecha de entrega: Hasta el domingo 21 de noviembre de 2021 a las 23:59.

Modo de entrega: A través de la tarea disponible en ModdleUA. Se podrá entregar tantas veces como se desee, sólo se corregirá la última entrega. Se debe entregar un único archivo comprimido `p2.tgz` o `p2.zip` creado en Linux con todos los ejercicios respetando para cada uno de ellos el nombre del enunciado.

IMPORTANTE: Puesto que se realizará una precorrección automática de la práctica, se deben respetar escrupulosamente tanto el formato de entrada como el de salida indicados en el enunciado.

Ejercicios:

1. (**p2-1.py**) Diseña un programa que lea una lista de números enteros separados por espacios en blanco y calcule:

- a) La media de los números leídos.
- b) La desviación típica muestral:

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \hat{x})^2}$$

- c) La moda (el valor que más veces aparece). Si hay dos o más que se repiten el mismo número de veces tu función debe devolver el primero que aparezca en el vector.

Debes utilizar una función para leer el vector de números, y otra por cada apartado. Se deben imprimir los valores redondeados a 5 cifras decimales si fuera necesario.

Ejemplo:

| Entrada | Salida |
|-----------|---------------------|
| 1 2 3 2 4 | 2.4 1.14018 2 |

2. (**p2-2.py**) Escribe una función que reciba dos vectores \vec{v} y \vec{w} de tamaño n (la función tendrá, por tanto, tres argumentos), y devuelva un entero que indique si un vector es *mayor*, *menor* o *igual* que el otro. Se considera que un vector \vec{v} es *mayor* que otro \vec{w} cuando la componente i -ésima de \vec{v} es mayor que la misma componente de \vec{w} y todas las componentes anteriores a i son iguales en ambos vectores.

Los siguientes ejemplos definen (informalmente) esta relación de orden entre vectores.

Ejemplos:

$\vec{v} = (1, 2, 3, 4, 5)$ y $\vec{w} = (1, 2, 3, 4, 4)$ 1 (\vec{v} es *mayor* que \vec{w})
 $\vec{v} = (1, 2, 3, 4, 5)$ y $\vec{w} = (1, 2, 3, 5, 4)$ -1 (\vec{v} es *menor* que \vec{w})
 $\vec{v} = (1, 1, 3, 4, 5)$ y $\vec{w} = (1, 1, 3, 4, 5)$ 0 (\vec{v} y \vec{w} son *iguales*)

Una vez hayas diseñado la función, haz un programa que lea un número entero n y dos vectores de n componentes cada uno (utilizando una función que lea un vector) y los compare utilizando la función, escribiendo el resultado por pantalla:

Ejemplo:

| Entrada | Salida |
|-----------------------------|--------|
| 5 1 2 3 4 5 1 2 3 5 4 | -1 |

3. (**p2-3.py**) Diseña un programa que lea un vector de componentes enteras y lo ordene de menor a mayor utilizando el siguiente método (conocido como *método de intercambio*):
 - a) Buscar dentro del vector el mínimo e intercambiarlo con el valor situado en la primera posición. Por ejemplo, si el vector original es $\vec{v} = (4, 7, 2, 1, 5)$, el mínimo es 1, y se intercambiaría con el 4, quedando $\vec{v} = (1, 7, 2, 4, 5)$.
 - b) Volver al paso anterior avanzando una posición, y buscando el mínimo a partir de la posición siguiente a la intercambiada. En el ejemplo, habría que buscar el mínimo en $(7, 2, 4, 5)$.

El proceso termina cuando se llega al final del vector (a la última posición). Una vez ordenado, el programa debe escribir el vector por pantalla. Debes utilizar funciones para leer, ordenar y escribir el vector.

Ejemplo:

| Entrada | Salida |
|---------------|---------------|
| 5 7 8 2 4 3 1 | 1 2 3 4 5 7 8 |

4. (**p2-4.py**) Diseña un programa que lea un vector de componentes enteras, un entero p que llamaremos posición, un entero l que llamaremos longitud y construya un nuevo vector con el trozo del vector original que empieza en la posición p y tiene longitud l . Antes, el programa debe comprobar $0 \leq p \leq n - 1$, (n es el tamaño del vector introducido), y que $p + l \leq n$. Si no se cumple alguna de estas dos condiciones el programa debe escribir "error".

Ejemplos:

| Entrada | Salida |
|-------------------------|-------------|
| 5 7 8 2 4 3 1 7 1 | error |
| 5 7 8 2 4 3 1 6 2 | error |
| 5 7 8 2 4 3 1 2 3 | 8 2 4 |
| 5 7 8 2 4 3 1 0 6 | 5 7 8 2 4 3 |

5. (**p2-5.py**) Diseña un programa que lea un entero n , y una matriz cuadrada de $n \times n$ componentes enteras, y a continuación calcule lo siguiente:
- La media de los elementos de la diagonal de la matriz,
 - El máximo de los elementos de la *diagonal opuesta*, y
 - Considerando las dos filas centrales como vectores, la distancia euclídea entre ambos vectores. Si n es par, las filas centrales son fáciles de calcular; si n es impar, sólo hay una fila central, luego se tomará la fila anterior como el otro vector.

Por ejemplo, si M fuese:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

habría que calcular la media de 1, 6, 11 y 16, el máximo de 4, 7, 10 y 13, y la distancia entre (5, 6, 7, 8) y (9, 10, 11, 12).

Debes hacer una función para leer la matriz, otra para calcular la media de la diagonal, otra para el máximo de la otra diagonal y finalmente otra función que calcule la distancia euclídea entre dos vectores.

Ejemplos:

| Entrada | Salida |
|--|-------------------|
| 4 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 | 8.5 13 8 |
| 3 1 4 7 2 5 8 3 6 9 | 5 7 1.73205 |

6. (p2-6.py) Escribe un programa que lea dos números enteros n y m y a continuación lea una matriz M de $n \times m$ valores enteros. Una vez leída la matriz M , debe obtener la matriz traspuesta M^T y escribirla, calcular el producto de M por su traspuesta, y escribir el resultado de dicho producto.

Ejemplo:

| Entrada | Salida |
|--|--|
| 3 4 1 4 7 10 2 5 8 11 3 6 9 12 | 1 2 3 4 5 6 7 8 9 10 11 12 166 188 210 188 214 240 210 240 270 |

7. (p2-7.py) Diseña un programa que:
- Lea dos enteros, n y m ,
 - Lea un vector \vec{v} de n componentes (reales),
 - Lea m vectores de n componentes cada uno y los guarde en una matriz.
 - Escriba por pantalla el vector de los m vectores que esté más cercano a \vec{v} , utilizando la distancia euclídea.

Ejemplo:

| Entrada | Salida |
|--|---------------|
| 3 4 1.1 2.2 3.3 1.05 1.99 3.2 7.08 4.56 3.21 9.21 3.2 6.5 0.96 8.7 2.1 | 1.05 1.99 3.2 |