

GUÍA PRÁCTICA

Ejercicios de programación

Licenciatura en Gestión de Tecnologías de la Información



Asignatura:

- Introducción a la Programación

1. Introdutorios

*Entradas / Salidas, Variables, Operadores y Estructuras de Decisión.

- 1.1. Escribir un programa que diga “Bienvenidos al Curso de Programación”
- 1.2. Escribir un programa que almacene la cadena ¡Hola Mundo! en una variable y luego muestre por pantalla el contenido de la variable.
- 1.3. Escribir un programa que pregunte el nombre del usuario en la consola y después de que el usuario lo introduzca muestre por pantalla la cadena ¡Hola <nombre>!, donde <nombre> es el nombre que el usuario haya introducido.
- 1.4. Ingresar dos números e informar la suma, multiplicación, división y resta de ellos. Si el segundo número es cero la división debe mostrar “No puede realizarse la operación”
- 1.5. Hacer un programa en el que se ingresa un número entero y por pantalla informa el doble de ese número.
- 1.6. Hacer un programa que ingresas la edad de una persona y calcula cuántos años va a tener dentro de 23 años. Mostrar el resultado por pantalla.
- 1.7. Hacer un programa en el que se ingresa precio y calcula el precio final con el iva (21%). Mostrar el resultado por pantalla
- 1.8. Hacer un programa que ingresa un precio y calcula un descuento del 9%. Mostrar por pantalla el resultado
- 1.9. Hacer un programa en el que ingresas un número que representa cierta cantidad de una fruta y divides esa cantidad por 3 chicos. Mostrar por pantalla el resultado
- 1.10. Pedir al usuario su edad y mostrar si es mayor o no de edad
- 1.11. Escribir un programa que pregunte al usuario por el número de horas trabajadas y el coste por hora. Después debe mostrar por pantalla la paga que le corresponde.
- 1.12. Hacer un algoritmo que nos permita introducir un número, luego muestre por pantalla el mensaje “Este número es positivo” o “Este número es negativo”
- 1.13. Hacer un algoritmo que nos permita introducir un número, luego muestre por pantalla el mensaje “Este número es par” o “este número es impar”
- 1.14. Hacer un algoritmo que pida un número entero y determine si es múltiplo de 5. Muestra un mensaje por pantalla.
- 1.15. Hacer un algoritmo que ingrese dos números y determinar si alguno de los dos números es divisible por 3.
- 1.16. Hacer un algoritmo que ingresa una letra, luego mostrarla por pantalla.
- 1.17. Hacer un algoritmo que ingresa 2 letras, mostrarlas por pantalla ordenadas alfabéticamente.

- 1.18. Hacer un algoritmo que ingresé un número decimal, súmale 15 y divídí por 2 mostrar el resultado por pantalla en formato decimal
- 1.19. Un señor tiene 10 pares de zapatillas, esa cantidad la ingresa a su computadora, en el momento del ingreso le llegan 5 pares más, que debe ingresar también en la computadora. Mostrar por pantalla cuantas zapatillas tiene en total
- 1.20. Hacer un algoritmo que ingresa un número, mostrar por pantalla si es par
- 1.21. Hacer un algoritmo que se ingresa un precio (número decimal) y le calcula un incremento de 10% Mostrar por pantalla el resultado
- 1.22. Hacer un algoritmo que se ingresa dos precios (número decimal) Mostrar los precios ordenado de mayor a menor
- 1.23. Hacer un algoritmo que se ingresa el tiempo registrado de dos autos que corrieron una carrera. El tiempo se guardó como número decimal. Mostrar por pantalla cual llegó primero
- 1.24. Hacer un algoritmo que se ingresa tres números. Mostrar por pantalla si están ordenados de menor a mayor
- 1.25. Hacer un algoritmo que permita ingresar tres números enteros y, si el primero de ellos es negativo, calcular el producto de los tres, en caso contrario calcular la suma de ellos.
- 1.26. Diseñar un algoritmo que calcule la superficie de un triángulo a partir del ingreso de su base y altura y muestre por pantalla el resultado, con el mensaje “La superficie del triángulo es:.”.
- 1.27. Realizar un algoritmo que permita leer dos números, determinar cuál de los dos números es el menor y mostrarlo por pantalla con el mensaje “El numero N es menor al número X”
- 1.28. Diseñar un algoritmo que calcule la longitud de la circunferencia y el área del círculo. Considerar que por pantalla se ingresa el radio de dicha circunferencia.
- 1.29. Realizar un algoritmo que permita determinar la hipotenusa de un triángulo rectángulo. Si se considera como dato las longitudes de sus dos catetos y esto se ingresa por pantalla.
- 1.30. A un trabajador le pagan según sus horas trabajadas y la tarifa está a un valor por hora. Si la cantidad de horas trabajadas es mayor a 40 horas, la tarifa por hora se incrementa en un 50% por considerar que son horas extras. Desarrollar un programa que calcule el salario del trabajador dadas las horas trabajadas y la tarifa
- 1.31. Diseñar el algoritmo que permita ingresar dos valores por ejemplo NUM1 y NUM2 y luego se intercambien los valores de dichas variables, es decir que el valor que tenía NUM1 ahora lo contenga NUM2 y viceversa
- 1.32. Ingresar por teclado 3 números correspondientes a los lados de un triángulo. Teniendo en cuenta que la suma de los dos lados menores tiene que ser superior al lado mayor para que formen un triángulo, indicar si los números indicados forman un triángulo y si lo forman que tipo de triángulo es (según sus lados).

- 1.33. Realizar un programa que solicite el valor hora de un empleado. Posteriormente se ingresa el nombre del empleado, la antigüedad y la cantidad de horas trabajadas en el mes. El programa debe calcular el importe a cobrar de la siguiente forma:
- El subtotal resulta de multiplicar el valor hora por la cantidad de horas trabajadas
 - Al subtotal hay que sumarle la cantidad de años trabajados multiplicados por \$30
 - A este resultado hay que restarle el 13% en concepto de descuentos.
- Mostrar por pantalla el recibo del empleado con el nombre, la antigüedad, el valor hora, el total a cobrar en bruto, el total de descuentos y el valor neto a cobrar.

2. Estructuras iteración (Bucles)

*Ciclos FOR, WHILE

- 2.1. Escribir un programa que pregunte el nombre del usuario en la consola y un número entero e imprima por pantalla en líneas distintas el nombre del usuario tantas veces como el número introducido.
- 2.2. Ingresar dos números y mostrar el menor de ellos. El ejercicio finaliza cuando se ingresan números iguales
- 2.3. Hacer un algoritmo que se ingresan 10 números enteros, mostrar por pantalla la cantidad de números pares que se ingresaron.
- 2.4. Escribir un programa que pida al usuario una palabra y la muestre por pantalla 10 veces. (realizarlo con while y con for)
- 2.5. Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla todos los números impares desde 1 hasta ese número separados por comas.
- 2.6. Escribir un programa que pida al usuario un número entero positivo y muestre por pantalla la cuenta atrás desde ese número hasta cero separados por comas.
- 2.7. Escribir un programa que pida al usuario un número entero y muestre por pantalla un triángulo rectángulo.

EJ: para 5

```
*
**
***
****
*****
```

2.8. Escribir la tabla de multiplicar de un número el cual se pide por pantalla.

Ej. Calcule la tabla del: 5

```
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
```

2.9. Dado un número natural X informar por pantalla su último dígito. En caso de no cargar un número informar un mensaje de error.

Realizarlo con método matemático y luego con el uso de Slicing

Ej: Ingresa: 124 Resultado: 4

2.10. Dado un número natural x, contar la cantidad de dígitos que posee.

Realizarlo mediante

- A) método matemático (división sucesiva de 10)
- B) Uso de la función len
- C) Uso del ciclo for.

2.11. Hacer un algoritmo donde se ingresan 10 números enteros. Mostrar por pantalla el número más grande ingresado y en qué posición se ingresó.

2.12. Escribir un programa que pida al usuario una palabra y luego muestre por pantalla una a una las letras de la palabra introducida empezando por la última.

Ejemplo. Ingresa: Hola

```
Ingresa palabra: hola
a
l
o
h
```

2.13. Escribir un programa en el que se pregunte al usuario por una frase y una letra, # y muestre por pantalla el número de veces que aparece la letra en la frase.

```
Ingresa una frase: capicua
Ingresa la letra: c
La letra ingresada aparece 2 veces
```

- 2.14. Dado un número natural x, contar la cantidad de dígitos pares e impares que posee.

```
Ingresar un número natural: 1587
La cantidad de dígitos pares son: 1
La cantidad de dígitos impares son: 3
```

- 2.15. Hacer un algoritmo que se ingrese 10 letras. Mostrar cuál es la mayor letra ingresada.
- 2.16. Hacer un algoritmo donde se ingresan los siguientes datos de 5 alumnos:
- nota (float),
 - Edad = (entero)
- Mostrar por pantalla la mejor nota y qué edad tenía.
- 2.17. Hacer un algoritmo que registra la temperatura (número decimal), todos los días a las 15 hs durante toda una semana cualquiera. Mostrar por pantalla la temperatura más alta de esa semana y la más baja.
- 2.18. Hacer un algoritmo que permita ingresar diez (10) números, luego muestre por pantalla cuántos eran mayores a cero y cuántos son menores a cero.
- 2.19. Hacer un algoritmo que permita ingresar por pantalla cinco (5) números y luego calcular su media.
- 2.20. Hacer un algoritmo que se ingresan 50 números enteros, calcular cuántos números impares se ingresaron
- 2.21. Hacer un algoritmo que se ingresan 20 números enteros, calcular cuántos números impares y cuántos números pares fueron ingresados
- 2.22. Hacer un algoritmo que pida la nota de un examen (un n° real entre 0 y 10) muestre por pantalla la calificación de la siguiente forma:
- Si la nota es menor que 5 la leyenda ***“En suspenso”***
 - Si la nota se encuentra entre 5 inclusive y 7 sin incluir la leyenda ***“Aprobado”***
 - Si la nota se encuentra entre 7 inclusive y 9 sin incluir la leyenda ***“Notable”***
 - Si la nota se encuentra entre 9 inclusive y 10 sin incluir la leyenda ***“Sobresaliente”***
 - Si la nota es un 10 la leyenda ***“Matrícula de honor”***.

Terminar el algoritmo cuando se ingresa cero como nota. En caso que no, vuelve a pedir una nota.

- 2.23. Hacer un algoritmo que permita ingresar por pantalla números naturales al finalizar informar:
- ¿Cuántos están entre el 50 y 75, ambos inclusive?
 - ¿Cuántos mayores de 80?
 - ¿Cuántos menores de 30?

El algoritmo debe finalizar cuando se ingresa el número 0.

- 2.24. Hacer un algoritmo que permita convertir calificaciones numéricas a letras, según la siguiente tabla:
- A=10
 - B= 8 y 9

- C=6 y 7
- D=4, y 5
- E=1 hasta el 4.

Se asume que la nota está comprendida entre 1 y 10. (En caso de ingresar números fuera de rango, informar con un mensaje “Rango nota invalido”

Informar por pantalla la calificación (Letra) que se sacó.

El programa finaliza cuando se ingresa el número cero o menor.

- 2.25. Hacer un algoritmo que permita ingresar un número natural luego por pantalla informar todos sus divisores.
- 2.26. Realizar un algoritmo para calcular el cuadrado de un número. El número debe ser mayor que cero. Si es menor que cero mostrar por pantalla el mensaje ***“ERROR, el Número debe ser mayor que cero o cero para finalizar”***. Si no mostrar el resultado de dicho cálculo. El programa finaliza cuando se ingresa cero.
- 2.27. Diseñar un algoritmo que permita ingresar por pantalla 10 números naturales. Calcular el promedio de los números pares. Luego mostrar por pantalla el resultado.

- 2.28. El algoritmo deberá pedir por pantalla un número entero natural. Calcular la Suma de los Primeros n Números Enteros utilizando ciclos iterativos.
- EJ: Si ingresa un 5 el algoritmo realiza
- $$1+2+3+4+5 = 15$$
- 2.29. Desarrollar un algoritmo que lea una lista de 20 números y determine cuántos son positivos, y cuantos son negativos.
- 2.30. Desarrollar un algoritmo que permita ingresar 50 números naturales, luego informar por pantalla:
- A. La sumatoria de los valores múltiplos de 3.
 - B. La cantidad de valores múltiplos de 5.
 - C. La sumatoria de los valores que se ingresan y son par.
- 2.31. Desarrollar un programa que permita ingresar números mayores que 0 (cero), finalizar el ingreso cuando no se cumpla esta condición e informar el valor menor y el valor mayor del conjunto.
- 2.32. Escriba un programa donde se reciba en primera instancia la cantidad de personas a encuestar. Luego de ello comienza el proceso indagando acerca del sexo, edad y altura. Se pide realizar un programa que informe el porcentaje de mujeres mayores de 25 años, la cantidad de varones menores de 18 años y el porcentaje de individuos mayores de 18 años cuya altura supera los 170 cm. Ingresar únicamente 40 encuestas.
- 2.33. Hacer un algoritmo donde se permita el ingreso de un número desde y numero hasta. A partir de allí realice la sumatoria de los números enteros comprendidos entre el desde y hasta. Agregar la validación de que el número desde no sea mayor al hasta
- 2.34. Realizar un programa que permita el ingreso de un número y determine si es primo o no. Validar el ingreso de números mayores a 0.
- 2.35. Desarrollar un algoritmo que permita el ingreso por pantalla de un número entero (x) y realice la sumatoria de los números enteros pares comprendidos entre el 1 y dicho número, es decir, $2+4+6+...+x$. El programa deberá mostrar por pantalla los números en cuestión con un mensaje y finalmente su sumatoria

- 2.36. Diseñar un algoritmo que permita ingresar 10 números, ninguno de ellos igual a cero. Se pide sumar los positivos, obtener el producto de los negativos y luego mostrar ambos resultados.
- 2.37. Hacer un programa que permita ingresar distintos caracteres el programa finaliza cuando han ingresado 10 veces la letra 'a'. Por cada carácter ingresado y que no sea una 'a' deberá mostrar un mensaje ***“No es la letra A”***. Cuando ingresaron 10 letras 'a' el programa terminó.
- 2.38. Hacer un programa que lea caracteres desde teclado y vaya contando las vocales que aparecen. El programa terminará cuando lea el carácter * (asterisco) y entonces mostrará un mensaje indicando cuántas vocales ha leído (cuántas de cada una de ellas).

- 2.39. Escribir un programa que ayude a aprender las tablas de multiplicar. Para ello se irá pidiendo la tabla de multiplicar de un número (pedido por teclado con anterioridad) y se comprobará que los valores introducidos sean correctos. Si es así el programa escribirá “correcto” y en caso contrario deberá escribir “Lo siento, se ha equivocado. La respuesta correcta era” La última línea mostrará el número de aciertos.

A continuación, se muestra un ejemplo de ejecución:

Programa para aprender las tablas de multiplicar

- Por pantalla se escribe la siguiente pregunta “¿Con qué número quieres practicar?” el usuario ingresa el número de la tabla que quiere practicar 5
- Luego mostrar por pantalla $5 * 1 =$
 - Dato introducido por el usuario 5
 - El programa envía el mensaje “Valor correcto”
- Luego mostramos por pantalla $5 * 2 =$
 - El usuario introduce 11
 - El programa envía el mensaje “Lo siento se ha equivocado. La respuesta correcta era 10”
- Cuando finalizamos le enviamos el siguiente mensaje “Has acertado 9 veces de 10”

- 2.40. Escribir un programa que primero pida por pantalla con cuántos números se va a trabajar (N). Luego se deberán ingresar N veces nuevos valores numéricos Y. El algoritmo deberá indicar cuál es el mayor y menor valor introducido, así como el promedio de todos los números introducidos (Y).

- 2.41. Escribir un programa que visualice una tabla con su cuadrado y cubo de los N primeros números, siendo N un

Número que ingresa el usuario. Utilizar el siguiente diseño de salida suponiendo que el Usuario ingresó un tres:

NÚMERO CUADRADO CUBO

1	1	1
2	4	8
3	9	27

- 2.42. Diseñar un algoritmo que permita calcular el promedio de un alumno sabiendo que los datos que se ingresan por alumno es:

Nombre del alumno

Nombre de la materia

Seis notas de esa materia,

Cuando en nota se ingresa cero se finaliza el programa. Se pide mostrar: **NOMBRE DEL ALUMNO:.....PROMEDIO:.....**

3. Estructuras de Datos Vectores (1D)

- 3.1. Realizar un programa que cargue un vector de 15 posiciones.
- 3.2. Desarrollar un algoritmo que permita la carga de un vector de 10 posiciones. Generar una rutina que transcriba el contenido del vector a otro vector en orden inverso.

a. Ejemplo dado: Vector Origen

1	1	2	2	3	4	6	7	9	5
---	---	---	---	---	---	---	---	---	---

b. Resultado Esperado: Vector Destino

5	9	7	6	4	3	2	2	1	1
---	---	---	---	---	---	---	---	---	---

- 3.3. Diseñe un programa que lea un vector de 10 posiciones, luego determine la primera posición que contenga un número múltiplo de 3. En caso que no lo contenga deberá informar por pantalla “No existe”

Ejemplo

1	1	2	2	6	4	6	7	9	5
---	---	---	---	---	---	---	---	---	---

Resultado Esperado: “El número múltiplo de tres es 6 y está en la posición 4 “

- 3.4. Crear un programa que lea e imprima el promedio del total de los elementos pares de un vector de 10 posiciones
- 3.5. Crear un programa con un vector de 5 posiciones que lea todas las posiciones y luego imprima el doble de los valores leídos en cada una de sus posiciones.
- 3.6. Crear un programa con un vector ORIGEN de 10 posiciones que lea todas las posiciones y luego traslade el doble de los valores en aquellas posiciones impares a otro vector DESTINO.

a. Ejemplo dado: Vector Origen

1	1	2	2	3	4	6	7	9	5
---	---	---	---	---	---	---	---	---	---

b. Resultado Esperado: Vector Destino

1	2	2	4	3	8	6	14	9	10
---	---	---	---	---	---	---	----	---	----

- 3.7. Desarrollar un algoritmo lea desde teclado un vector de 10 posiciones y que detecte los elementos repetidos de un vector y los reemplace con un 0 y los deposite en otro vector.

Debe mostrarse por pantalla el vector resultante

Ejemplo dado:

2	1	2	3	4	11	3	2	5	9
---	---	---	---	---	----	---	---	---	---

Resultado Esperado:

0	1	0	0	4	11	0	0	5	9
---	---	---	---	---	----	---	---	---	---

- 3.8. Dado un vector de 10 posiciones se debe reemplazar los valores impares con una 0

Ejemplo dado:

2	1	2	3	4	11	3	2	5	9
---	---	---	---	---	----	---	---	---	---

Resultado Esperado

2	0	2	0	4	0	0	2	0	0
---	---	---	---	---	---	---	---	---	---

- 3.9. Dado un vector de 6 posiciones se debe

- Sumar los valores pares
- Multiplicar los valores impares

Mostrar el resultado por pantalla.


Ejemplo dado:

2	2	2	3	1	2
---	---	---	---	---	---


Resultado Esperado: Suma: 6 Multiplicación: 12

- 3.10. Escribir un programa que lea diez números, los guarde en un vector y a continuación los muestre por pantalla en orden inverso al de su entrada

- 3.11. Escribir un programa que lea tres números y los guarde en un vector. A continuación, los ordenará y guardará los valores ordenados en otro vector. Finalmente sacará ambos vectores por la pantalla
- 3.12. Realizar un programa que genere 3 vectores de 10 posiciones cada uno. El primero está formado por los números del 1 al 10. El segundo por el cuadrado de estos números y el tercero por el cubo.
- 3.13. Realizar un programa que por medio de un menú de opciones (usar la instrucción elif) y trabajando con un vector de 10 números reales me permita:
- a. Cargar el vector
 - b. Mostrar por pantalla el vector
- 3.14. Escribir un programa que tras asignar los números, -2, 5, 8, -9, 10, 15 y -4 a un arreglo calcule, independientemente, la suma de los elementos positivos y negativos.
- 3.15. Realizar un programa que genere 3 vectores de 10 posiciones cada uno. El primero está formado por los números del 1 al 10. El segundo por el cuadrado de estos números y el tercero por el cubo.
- 3.16. Realizar un programa que por medio de un menú de opciones (usar la instrucción elif y trabajando con un vector de 10 números reales me permita:
- A) Cargar el vector
 - B) Mostrar por pantalla el vector ordenado

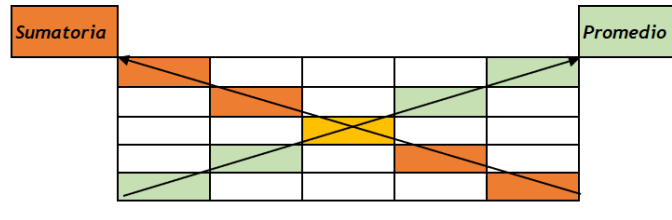
 Importante: Debe haberse dado tema “metodos de ordenamiento”

- 3.17. Hacer un programa que permita leer 20 números enteros (positivos y negativos) distintos de cero. Mostrar el vector tal como fue ingresado y luego mostrar los positivos ordenados en forma decreciente y por último mostrar los negativos ordenados en forma creciente.

 Importante: Debe haberse dado tema “metodos de ordenamiento”

4. Estructuras de Datos Matrices (2D)

- 4.1. Efectuar la carga de una matriz de 3x3 desde el teclado. Efectuar la sumatoria de valores alojados en una de sus diagonales y el promedio de los valores de la diagonal siguiente.



- 4.2. Efectuar la carga de una matriz de 5x5 desde el teclado. Efectuar la sumatoria de los valores alojados en la diagonal principal y el promedio de los valores alojados en la contradiagonal. Mostrar por pantalla los resultados.
- 4.3. Efectuar la carga de una matriz de 3x2 desde teclado. Obtener como resultado la matriz transpuesta. Mostrarla por pantalla.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^t = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^t = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

- 4.4. Efectuar la carga de dos matrices de 3x3 desde teclado. Obtener como resultado el producto de matrices y colocarlo en una nueva matriz. Mostrar el resultado por pantalla.

A	B	Resultado																											
<table border="1"><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>3</td><td>1</td><td>4</td></tr></table>	1	2	1	1	1	1	3	1	4	<table border="1"><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>3</td><td>1</td><td>1</td></tr></table>	1	2	1	1	1	1	3	1	1	<table border="1"><tr><td>1</td><td>4</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>9</td><td>1</td><td>8</td></tr></table>	1	4	1	1	1	1	9	1	8
1	2	1																											
1	1	1																											
3	1	4																											
1	2	1																											
1	1	1																											
3	1	1																											
1	4	1																											
1	1	1																											
9	1	8																											

- 4.5. Desarrollar una matriz de 3X3. Ingresar números enteros. Solicitar al usuario que ingrese un número de fila y obtener el número mayor ingresado en dicha fila.

- 4.6. Diseñar un algoritmo que permita la carga de una matriz de 3x3 desde teclado. A esta matriz la llamaremos MatOrigen.
Una vez cargada la matriz efectuar el análisis de los valores cargados y obtener como resultado otra matriz la cual llamaremos MatDestino en donde como resultado deberá alojar en la misma fila columna el valor 1 (uno) en caso de que el valor origen sea par y un 0 (cero) en caso de que el valor sea impar.
Mostrar el resultado de la matriz destino por pantalla.

Ver ejemplo ilustrativo.

MatOrigen

1	2	5
8	3	5
3	4	4

MatDestino

1	0	1
0	1	1
1	0	0

- 4.7. Efectuar un programa en C el cual permita la carga de una matriz de 3x3 desde teclado. A esta matriz la llamaremos MatOrigen.
Una vez cargada la matriz, obtener como resultado un vector de 2 posiciones al que llamaremos VecDestino, el cual deberá alojar en la posición 0 la SUMA de los valores origen Impares y en la posición 1 la suma de los valores origen pares.
Mostrar el resultado de Vecdestino por pantalla.
Ver ejemplo ilustrativo.

MatOrigen

1	2	5
8	3	5
3	4	4

VecDestino

17	18
----	----

- 4.8. Dada una matriz de 3X4, desarrollar un programa que solicite al usuario el ingreso de números enteros. Al finalizar se obtiene la suma de todos los números positivos ingresados y la suma de los números negativos.
- 4.9. Desarrollar una matriz de 3X3, ingresar números enteros. Al finalizar mostrar el mayor número ingresado y su posición.

- 4.10. Cada semana, un negocio de electrodomésticos registra las ventas de los productos individuales que hay en existencia. Al finalizar el mes, se genera un reporte semanal y se analizan. Un ejemplo de un mes típico se muestra en la siguiente tabla.

Semana	Televisor	Heladera	Lavarropa
1	3	4	3
2	5	6	5
3	3	2	1
4	6	7	8

Determinar:

- Calcular el número total de artículos vendidos cada semana
 - Calcular el número total de artículos vendidos en el mes.
- 4.11. Dada una matriz de 3x3 en la que se leen desde teclado valores numéricos y se lee número de fila o columna, se pide tener un menú principal. las operaciones

1) FILA 2) COLUMNA

1) SUMA 2) RESTA

Ejemplo dado:

1	1	3
2	3	2
3	5	1

- Se lee desde teclado 1) Fila
- Se lee desde teclado Número de Fila : 2
- Se lee desde teclado Operación matemática 1) Suma

Resultado Esperado: 7

5. Funciones

- 5.1. Realiza un programa principal que lea tres números enteros por teclado, los almacene en tres variables (x, y, z) y llame a una función llamada máximo(), con tres argumentos, que devuelva el máximo de estos tres valores.
- 5.2. Realiza un programa que permita el ingreso de dos cadenas de caracteres llame a unas funciones programadas por ti que realicen las mismas tareas que la función len() y strcmp() (Existentes en lenguaje C.)

len(str) -> myUDFcuantaCaracteres()
strcmp() -> myUDFcomparaCadenas()

- 5.3. Realiza un programa principal que lea una palabra desde el teclado e identifica si la misma es palindroma (capicua).
- 5.4. Realizar un programa principal que mediante el llamado de una función calcule el factorial de un número entero n.
 - A. Mediante una estructura repetitiva
 - B. Mediante la siguiente fórmula recursiva: $n! = n * (n-1)!$ Teniendo en cuenta que: $1! = 1$.

Por medio de un menú se podrá elegir si el resultado se calcula por la opción A o por la opción B.

- 5.5. Realizar funciones que mediante algoritmia propia emule el funcionamiento y propósito de los siguientes métodos de la librería array.

5.5.1.	append()	Adds an element at the end of the list
5.5.2.	clear()	Removes all the elements from the list
5.5.3.	copy()	Returns a copy of the list
5.5.4.	count()	Returns the number of elements with the specified value
5.5.5.	index()	Returns the index of the first element with the specified value
5.5.6.	insert()	Adds an element at the specified position
5.5.7.	pop()	Removes the element at the specified position
5.5.8.	remove()	Removes the first item with the specified value
5.5.9.	reverse()	Reverses the order of the list

- 5.6. Programe un método recursivo que calcule la suma de un arreglo de números enteros.
- 5.7. Programe un método recursivo que invierta los números de un arreglo de enteros.