

# TEORÍA DE LA PROGRAMACIÓN

DOCENTE: Wilman Chamba Zaragocín

## ACTIVIDADES DE APRENDIZAJE y PRÁCTICAS (USO DE PROGRAMACIÓN MODULAR Y ESTRUCTURAS DE DATOS ESTÁTICAS COMPUESTAS)

Los siguientes ejercicios tienen como finalidad que los estudiantes tengan la pericia y entrenamiento necesario para resolver problemas relacionados con las estructuras básicas, condicionales, repetitivas, estructuras de datos estáticas compuestas y programación modular que se utilizan en los algoritmos, como también inculcar la auto educación, auto evaluación, investigación y sobre todo la persistencia en resolución de problemas.

### INSTRUCCIONES:

Elaborar algoritmos y programas con funciones incluyendo todo el proceso de resolución de problemas a través de algoritmos computacionales. Deben encontrar: REQUERIMIENTO, ANÁLISIS Y DISEÑO, PSEUDOCÓDIGO, DIAGRAMA DE FLUJO, y CODIFICACIÓN EN LENGUAJE DE PROGRAMACIÓN C y EJECUCIÓN DE LOS PROGRAMAS. Para los mensajes de salida de sus algoritmos, póngase en el papel de usuario final para que muestre información oportuna. Además identifique controles de validación en algoritmos y/o programas donde sea necesario.

### Criterios para la entrega:

- Informe: Documento en PDF, donde se indique REQUERIMIENTO, ANÁLISIS Y DISEÑO, capturas de pantalla de: PSEUDOCÓDIGO DE FUNCIONES O MÓDULOS, CORRIDA o PRUEBA DE ESCRITORIO (3 casos como mínimo) CODIFICACIÓN EN LENGUAJE DE PROGRAMACIÓN C y EJECUCIÓN DEL PROGRAMA (3 como mínimo).
- Comentarios de autoría en los archivos fuentes (como en C). Cabe Recalcar que se deben hacer llegar los fuentes de los archivos generados.
- Presentar en ejecución: el título del algoritmo, nombres de autoría, ciclo y paralelo (ya sea en corrida o prueba de escritorio en caso de que lo haya relizado en Pseint y obligatorio en C)

### PROBLEMAS

1. Ingresar una cadena por teclado y permita indicar el número total de vocales que existen en la cadena.
2. Ingresar una cadena por teclado y permita indicar el número detallado de vocales que existen en la cadena y su respectivo total.
3. Ingresar una cadena por teclado y permita transformar a mayúsculas las letras que son necesarias.
4. Ingresar una cadena por teclado y permita transformar a minúsculas las letras que son necesarias.
5. Ingresar una cadena por teclado y permita eliminar los caracteres en blanco de la cadena.
6. Ingresar una cadena por teclado y me permita obtener una subcadena ingresando su posición inicial y el número de caracteres a obtener.
7. Ingresar dos cadena por teclado y me permita concatenar esas cadenas en una cadena resultante
8. Ingresar una cadena (oración) por teclado y permita obtener las palabras que se encuentran en cada oración.
9. Ingresar una cadena y la encripte sumando 3 al código ASCII de cada carácter. Mostrar en pantalla la cadena encriptada.
10. Ingresar una cadena y que se invierta el resultado (la muestre al revés)
11. Llenar un arreglo de longitud 20 con valores aleatorios e indicar cual es(son) el(los) número(s) mayor(es) y menor(es) generados.

12. Realizar un programa que me permita ingresar el nombre de estudiante y el ingreso de sus notas finales, mostrar sus notas con su respectivo promedio, nota mayor y nota menor.
13. Realizar la multiplicación de dos vectores ingresados desde el teclado, además debe: mostrar los vectores ingresados y el resultante, indicar cuál es el número mayor del vector resultante y la posición en que se encuentra, indicar cuál es el número menor del vector resultante y la posición en que se encuentra. Utilizar el paradigma de la programación modular a través del uso de funciones diseñadas por el estudiante.
14. Llenar un arreglo de longitud 20 con valores aleatorios y me permita buscar un número dentro de ese arreglo, en caso de encontrarlo devolver la posición, en caso de que no devolver un valor negativo (-1).
15. Ingresar el límite del arreglo para poder ingresar sus elementos desde teclado (valores positivos y negativos) y ordenarlo en forma ascendente.
16. Ingresar el límite del arreglo para poder ingresar sus elementos desde teclado (valores positivos y negativos) y ordenarlo en forma descendente.
17. Ingresar los límites de una matriz, llenar con valores aleatorios para cada fila y luego mostrar dicha matriz.
18. Ingresar los límites de una matriz, llenar con valores aleatorios para cada fila y luego mostrar dicha matriz en forma normal y en forma descendente a través de filas y columnas. Ejemplo

Límite filas: 3

Límite columnas: 4

Matriz Original

1	2	10	20
3	5	7	9
8	2	0	67

Matriz Descendente

67	0	2	8
9	7	5	3
20	10	2	1

19. Ingresar los límites de una matriz, llenar con valores aleatorios para cada fila y luego mostrar dicha matriz en forma ordenada por sus valores.
20. Llenar una matriz con tabla de multiplicar universal (PITAGÓRICA) hasta un límite deseado y luego mostrar la respectiva matriz. Ejemplo: tabla de multiplicar hasta el 4, su salida será
 

X	1	2	3	4
1	1	2	3	4
2	2	4	6	8
3	3	6	9	12
4	4	8	12	16
21. Sumar dos matrices ingresadas por parte de la usuario y presentar las matrices ingresadas y la matriz resultante.
22. Multiplicar dos matrices ingresadas por parte de la usuario y presentar las matrices ingresadas y la matriz resultante.
23. Ingresar dos matrices cuadradas por el teclado y calcular la matriz resultante de acuerdo a los siguientes criterios:
  - La diagonal principal es el resultado de multiplicar cada uno de los elementos de la diagonal principal de las matrices ingresadas.
  - Los elementos que se encuentran sobre la diagonal principal se suman cada uno de los elementos sobre la diagonal principal de las matrices ingresadas.

- Los elementos que se encuentran debajo a la diagonal principal se restan cada uno de los elementos debajo de la diagonal principal de las matrices ingresadas.

Ejemplo:

A			B		
4	8	7	3	1	6
0	6	8	6	7	9
9	0	5	0	2	3

RTA		
12	9	13
-6	42	17
-9	-2	15

24. Realizar el juego de tres en raya, indicando para un jugador la X y para el otro jugador O y finalizar el juego cuando exista un ganador.
25. Calcular la determinante de una matriz ingresada desde el teclado

**NOTA:** Es aconsejable que todos estos algoritmos sean traducidos a un lenguaje de programación (en este caso Lenguaje C) como práctica de traducción de algoritmos en un lenguaje de programación