



EXAMEN DE TEORÍA DE LA PROGRAMACIÓN
EVALUACIÓN RECUPERACIÓN

NOMBRES Y APELLIDOS: _____ **CALF.** _____
FECHA: 2019 _____ **1 MÓDULO “ ”** **Wilman Chamba Z. (2 HORAS)**

RECUERDE: La evaluación es personal, estudiante que se encuentre copiando o en el intento se le retirará la evaluación y NO poseerá calificación. Favor lea detenidamente las preguntas. El motivo de esta prueba es validar sus conocimientos y sobre todo su lógica en la Teoría de la programación, para que en un futuro pueda seguir con sus estudios...ÉXITOS

INSTRUCCIONES GENERALES:

- Crear un programas en C de acuerdo a cada instrucción en particular.
- En cada programa como en la ejecución de las mismas deberá existir los comentarios con lo siguiente: Nombres completos del Alumno, Fecha, Evaluación Recuperación, y toda esta información debe salir en las ejecuciones de los programas.
- Se debe entregar los siguientes archivos en la Terea denominada Evaluación Recuperación - Práctica del Entorno Virtual de aprendizaje (EVA o Classroom):
 - Documento en PDF con: [1] sus datos informativos; [2] Códigos en programa C , [3] las captura de pantallas de la ejecución del (los) programa(s) funcionando tanto de la parte obligatoria u opcional; y, [4] alguna información adicional solicitada en cada ejercicio.
 - Código(s) fuente(s) de (los) programa(s) de acuerdo al nombre indicado en cada ejercicio (?????.c)
- Tomar en cuenta las buenas normas de programación en C o PROGRAMACIÓN LIMPIA.

RUBRICA GENERAL:

- 1) Presentación de informe - ordenado, claro, correcto (0.50 puntos)
- 2) Si no cumple con todos estos requisitos afectará a su acreditación.

I. PARTE PRÁCTICA (6 PUNTOS)

PROGRAMA 1 (3.0 PUNTOS):

Escribir un algoritmo completo para el lenguaje de programación en C con uso de funciones que permita ingresar una matriz, presentar la matriz y comprobar si dicha matriz ingresada es una MATRIZ SIMÉTRICA.

Una matriz se considera SIMÉTRICA si cumple exactamente con:

1. Si la matriz es cuadrada.
2. Si presentan simetría con respecto a la diagonal principal, es decir si $M[i, j] == M[j, 1]$. A continuación se indica un ejemplo de Matriz

$$M = \begin{matrix} & \begin{matrix} 1 & 9 & 3 \end{matrix} \\ \begin{matrix} 9 \\ 3 \end{matrix} & \begin{matrix} 2 \\ -1 \end{matrix} & \begin{matrix} -1 \\ 5 \end{matrix} \end{matrix}$$

Para facilidad realizarlo en funciones de operatividad para: leer todos los elementos de la matriz, presentar la matriz, comprobar si la MATRIZ ES MATRIZ SIMÉTRICA.

INSTRUCCIONES: Crear un programa en C denominado VerificarMatrizSimétrica.c.

RÚBRICA:

- Implementación de función y llamado para ingresar la dimensión de la matriz (teniendo en cuenta la características de matriz cuadrada) (0.1)
- Implementación de función para ingresar los elementos de la matriz (0.3)
- Llamada función ingresar los elementos de la matriz (0.1)
- Implementación de función para visualizar la matriz (0.3)
- Llamada a función para visualizar la matriz (0.1);
- Implementación de la función para comprobar si la MATRIZ ES SIMETRICA (1.0).
- Llamada a la función comprobar si la MATRIZ ES SIMETRICA (0.1).
- Mensaje de comprobación MATRIZ ES SIMETRICA (0.2)
- Ejecución del programa correctamente (capturas de pantallas de 3 casos complejos) (0.8)



- Implementación de función y llamado para ingresar el orden de la matriz (teniendo en cuenta la características de matriz cuadrada) (0.25)
- Implementación de función y llamado ingresar la matriz (0.5)
- Implementación de función y llamado para visualizar la matriz (0.50)
- Implementación de función y llamado y mensaje de: comprobar si la MATRIZ ES SIMETRICA (1.0 puntos).
- Ejecución del programa correctamente (capturas de pantallas de 3 casos complejos) (1.0 punto)

PROGRAMA 2 (2.5 PUNTOS):

Escribir un algoritmo completo para el lenguaje de programación en C que permita ingresar el limite de un arreglo con sus elementos y ordenarlo ascendentemente, de tal manera, que la salida del programa debe presentar el arreglo original y posteriormente el arreglo ordenado ascendentemente.

Para ordenar el arreglo puedo utilizar el método de la burbuja el cual consiste en lo siguiente:

En comparar cada elemento de un array que va ha ser ordenado con el siguiente, y si están desordenados intercambiarlos, se realizan las cantidad de iteraciones necesarias hasta tener todos los elementos ordenados.

Ejemplo del método considerando el siguiente arreglo inicial:

50	26	7	9
----	----	---	---

En la primera iteración ocurriría lo siguiente: el elemento 0 del arreglo (que inicialmente es 50) debe compararse con todos los elementos del arreglo

26	50	7	9	Se intercambia 50 y 26
7	50	26	9	Se intercambia 26 y 7
7	50	26	9	No existe Intercambio

En la segunda iteración ocurriría lo siguiente: el elemento 1 del arreglo (que inicialmente es 50 de acuerdo a la última fila de la iteración anterior) debe compararse con los siguientes elementos del arreglo

7	26	50	9	Se intercambia 50 y 26
7	9	50	26	Se intercambia 9 y 26

En la tercera iteración ocurriría lo siguiente: el elemento 2 del arreglo (que inicialmente es 50 de acuerdo a la última fila de la iteración anterior) debe compararse con los siguientes elementos del arreglo

7	9	26	50	Se intercambia 50 y 26
---	---	-----------	----	------------------------

En la cuarta iteración no existe que comparar ya que el elemento 3 del arreglo (que inicialmente es 50 de acuerdo a la última fila de la iteración anterior) ya no existen más elementos en el arreglo por lo tanto EL ARREGLO YA QUEDARÍA ORDENADO DE LA SIGUIENTE MANERA:

7	9	26	50
---	---	----	----

INSTRUCCIONES: Crear un programa en C denominado OrdenarArreglo.c.

RÚBRICA:

- Implementación de función y llamado para ingresar el limite del arreglo) (0.1)
- Implementación de función para ingresar los elementos del arreglo (0.3)
- Llamada función ingresar los elementos del arreglo (0.1)
- Implementación de función para visualizar el arreglo (0.3)
- Llamadas a función para visualizar los arreglos original y ordenado (0.2);
- Implementación de la función para ordenar el arreglo (0.8).
- Llamada a la función para ordenar el arreglo (0.2).
- Ejecución del programa correctamente (capturas de pantallas de 3 casos complejos) (0.5)

ALGORITMO 3 (2.5 PUNTOS - OPCIONAL):



Escribir un algoritmo y programa que me permita la edad exacta de una persona a partir de la fecha de su cumpleaños y la fecha actual.

Por ejemplo: (Suponer que la fecha actual para el ejemplo es 2019-02-28)

Ingrese la fecha de su cumpleaños: 2000-01-20

Su edad exacta es: 19 año(s), 1 mes(es), 8 día(s)

RÚBRICA:

- Uso de struct FechaNacimiento: 0.2
- Implementación de función para obtener la struct FechaNacimiento a partir del ingreso de una cadena tal como se indica en el ejemplo: (0.5)
- Llamada a la función para obtener la struct FechaNacimiento (0.2)
- Implementación de la función para obtener la edad exacta de una persona (Esta función debe retornar una cadena de la siguiente manera y de acuerdo al ejemplo: 19 año(s), 1 mes(es), 8 día(s))(0.5)
- Llamada función para obtener la edad exacta de una persona (0.2)
- Presentación del mensaje correctamente como : Su edad exacta es: (llamada a función) (0.2)
- Ejecución del programa correctamente (capturas de pantallas de 3 casos) (0.7)