

LABORATORIO FINAL: Ciclos y algo más

Materia: Algoritmia y programación I

Profesora: Rocio Ramos Rodríguez

Monitores: Sebastián Hernández, Nicolás Buitrago y Juan David Solano Silva

Curso: IST2088_51 – 52 - 53

Fecha de entrega: viernes 5 de junio de 2020 hasta la 12:30 p.m.

Fecha de sustentación: sábado 6 de junio de 2020, hora por definir.

Enunciado:

Diseñar un Menú iterativo con las siguientes opciones:

1. **Triqui:** Se cuenta con 9 variables que representan las nueve casillas de un tablero de triqui. Todas las posiciones inician vacías (valor inicial = 0). Los turnos impares serán para el círculo 0 (valor=1), los turnos pares serán para la equis X(valor=2). El programa leerá en qué posición desea jugar el jugador actual. No se podrá reescribir una posición que ya tenga un valor, por lo tanto, el programa leerá posiciones hasta que se digite una posición válida (vacía). Una vez se lea una posición válida, se verificará si existe algún ganador.

Numeración de las Posiciones donde irían las variables

1	2	3
4	5	6
7	8	9

Se jugará el juego hasta que exista un ganador o hasta que se hayan realizado 9 jugadas. El ganador será quien tenga tres valores similares en línea. En el tablero de ejemplo se puede ver como el jugador 2 (X) ha ganado el juego porque tiene 3 figuras iguales en línea. Es decir, que las posiciones 3, 5 y 7 tienen el mismo valor y este es 2. Si al llegar a la jugada 9 no ha habido ningún ganador, entonces el juego se declara en empate.

Esta debe ser la salida:

O	O	X
O	X	
X		

Esta debe ser la implementación:

1	1	2
1	2	0
2	0	0

2. **Punto y fama:** El jugador 1 (PC) genera aleatoriamente un número de cuatro cifras, no debe tener dígitos repetidos (se debe validar que el número digitado no tenga repetidos, en este caso se debe solicitar un nuevo número y volver a hacer la validación hasta que se digite un número válido). Posteriormente el jugador 2 debe introducir combinaciones, no debe tener dígitos repetidos (se debe validar que el número digitado no tenga repetidos, en este caso se debe solicitar un nuevo número y volver a hacer la validación hasta que se digite un número válido), hasta acertar el número original introducido por el jugador 1. Luego que el jugador 2 escriba cada combinación, se debe mostrar el número de aciertos obtenidos (se dice el número de famas (dígitos en la misma posición) y el número de puntos (dígitos en el número original pero no en la misma posición)). Al final, mostrar el número de intentos empleados en hallar el número.

3. **Funciones trigonométricas con series de Taylor:** Teniendo en cuenta que el seno y coseno se pueden calcular utilizando las siguientes fórmulas o series de Taylor, el programa debe permitir calcular las demás funciones basadas en estas 2. Tener en cuenta que la sumatoria debe tener un fin para poder dar un resultado.

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}, \forall x$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}, \forall x$$

4. **Clave de un número:** Elaborar un algoritmo que calcule la clave para un número entero, así, si el número es negativo se debe escribir la clave es -1. Si el número es positivo se calcula la clave de la siguiente forma: El dígito más a la derecha se multiplica por 2, el siguiente hacia la izquierda por tres, el siguiente hacia la izquierda por 4, etc. A la suma de los resultados de las multiplicaciones, se les calcula el mod 10. Por ejemplo si el número es 31234

$$4 \cdot 2 = 8$$

$$3 \cdot 3 = 9$$

$$2 \cdot 4 = 8$$

$$1 \cdot 5 = 5$$

$$3 \cdot 6 = 18$$

$$\text{Suma} = 18 + 5 + 8 + 9 + 8 = 48$$

$$\text{Mod} = 48 \bmod 10 = 8$$

5. **Multiplicación Rusa:** Diseñar un algoritmo que multiplique dos números enteros mediante el algoritmo de la multiplicación rusa. Este algoritmo multiplica por 2 el multiplicando y divide por dos el multiplicador hasta que el multiplicador toma el valor 1. Después suma todos los multiplicandos correspondientes a multiplicadores impares. Dicha suma es el producto de los números originales.

Ejemplo: $37 \cdot 12 = 444$ (multiplicador: 37, multiplicando: 12)

37	12
18	24
9	48
4	96
2	192
1	384

$$12 + 48 + 384 = 444$$

INSTRUCCIONES:

1. La interfaz debe ser muy amigable y visualmente agradable al usuario.
2. La sustentación equivale al 60% del laboratorio y el programa enviado al link correspondiente del catálogo al 40%
3. El laboratorio debe ser presentado en equipos de 3 integrantes.
4. El programa debe quedar guardado con el nombre Lab_Final_nombre-apellido1_nombre-apellido2_nombre-apellido3 y debe enviar uno solo de los integrantes el proyecto comprimido al link correspondiente en el catálogo.
5. El intento de copia será calificado con 0,0 y anotación a la hoja de vida.
6. Bono para utilizar en examen final: Usar JOption Pane para todas las entradas y salidas y además enviar el ejecutable del programa o proyecto en Netbeans.