	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de programación

Grupo: 20

No. de práctica(s): 03

Integrante(s): Juan Manuel Cuellar Orbezo

No. de lista o brigada: 04

Semestre: 2024-2

Fecha de entrega: 27 de febrero del 2024

Observaciones: -

CALIFICACIÓN: _____

Práctica 3 - Solución de problemas y Algoritmos

Objetivo

El alumno elaborará algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

Desarrollo

Algoritmos

Un algoritmo es como una receta para resolver un problema o realizar una tarea. Da instrucciones precisas, paso a paso, que se pueden seguir para llegar a un resultado final.

Los algoritmos se usan en todo tipo de cosas, desde matemáticas y ciencias hasta computadoras e incluso en la vida diaria. Por ejemplo, cuando se sigue una receta de cocina, se está usando un algoritmo. O al buscar una palabra en un diccionario, también se está usando un algoritmo.

Para que un algoritmo funcione bien, debe cumplir con algunas características importantes:

- **Precisión:** Las instrucciones deben ser claras y sin ambigüedad, indicando el orden exacto en que se deben realizar los pasos.
- **Definido:** Si se ejecuta el algoritmo varias veces con los mismos datos, siempre se debe obtener el mismo resultado.
- **Finito:** El algoritmo debe tener un número determinado de pasos, no puede ser infinito.
- **Correcto:** Debe cumplir con el objetivo para el que fue creado y dar la respuesta correcta.
- **Salida:** Debe tener al menos una salida que sea perceptible, es decir, que se pueda observar o entender.
- **Simplicidad:** Debe ser fácil de entender y seguir, sin pasos innecesariamente complejos.
- **Eficiencia:** Debe ser capaz de realizar la tarea en el menor tiempo posible, utilizando los recursos disponibles de forma eficiente.
- **Eficacia:** Debe producir el efecto deseado y cumplir con el objetivo para el que fue diseñado.

En resumen, un algoritmo es un conjunto de instrucciones precisas que ayudan a resolver un problema o realizar una tarea de forma eficiente.

En clase, para esta práctica, elaboramos varios ejemplos de algoritmos en pseudocódigo para demostrar nuestra comprensión de los diferentes tipos de algoritmos y las estructuras de control básicas, así como para desarrollar nuestra capacidad para implementar algoritmos de manera eficiente y legible.

Ejercicio 1

PROBLEMA: Determinar si un número dado es positivo o negativo.
RESTRICCIONES: El número no puede ser cero.

DATOS DE ENTRADA: Número real.
DATOS DE SALIDA: La indicación de si el número es positivo o negativo
DOMINIO: Todos los números reales.

1. Solicitar al usuario un número y almacenarlo como NUM. Ir al paso 02.
2. Analizar si NUM = 0. En caso de ser VERDADERO ir al paso 03 de lo contrario ir al paso 04.
3. Mostrar en pantalla “El número seleccionado no puede ser 0” Ir al paso 01.
4. Si NUM < 0 mostrar en pantalla “El número seleccionado es negativo”. De lo contrario Ir al paso 05
5. Mostrar en pantalla “El número seleccionado es positivo”. Ir al paso 06
6. Fin del algoritmo

Iteración	NUM	Salida
1	-5	El número seleccionado es negativo
2	67	El número seleccionado es positivo
3	0	El número seleccionado no puede ser 0

Tabla. 01 – Prueba de escritorio del ejercicio 1. Nótese como el algoritmo no deja que el valor seleccionado sea 0.

Ejercicio 2

PROBLEMA: Obtener el mayor de dos números dados.
RESTRICCIONES: Los números de entrada deben ser diferentes.

DATOS DE ENTRADA: Dos números reales.
DATOS DE SALIDA: La escritura del número más grande.

DOMINIO: Todos los números reales.

1. Solicitar al usuario un número y almacenarlo como NUMA. Ir al paso 02.

2. Solicitar al usuario un número y almacenarlo como NUMB. Ir al paso 03
3. Analizar si NUMA = NUMB. En caso de ser VERDADERO ir al paso 04 de lo contrario ir al paso 05.
4. Mostrar en pantalla “Ambos números no pueden ser iguales.” Ir al paso 01.
5. Si NUMA < NUMB mostrar en pantalla “El número mayor es {NUMB}”. De lo contrario Ir al paso 06
6. Mostrar en pantalla “El número mayor es {NUMA}. Ir al paso 07
7. Fin del algoritmo

Iteración	NUMA	NUMB	Salida
1	15	730	El número mayor es 730.
2	3.1416	0	El número mayor es 3.1416.
3	54	54	Ambos números no pueden ser iguales.

Tabla. 02 – Prueba de escritorio del ejercicio 2. Nótese como el algoritmo no deja que ambos valores sean iguales.

Ejercicio 3

PROBLEMA: Convertir grados Celsius a Fahrenheit
RESTRICCIONES: Número real.

DATOS DE ENTRADA: Un número real.
DATOS DE SALIDA: La temperatura en ℉.

DOMINIO: Todos los números reales.

1. Crear una variable que se llame TEMPF y asignarle el valor TEMPF = 0. Ir al paso 02.
2. Solicitar al usuario un número y almacenarlo como TEMPC. Ir al paso 03.
3. Obtener el valor de TEMPF a partir de la fórmula: TEMPF = (TEMPC * 1.8) + 32. Almacenar este valor numérico como TEMPF. Ir al paso 04.
4. Mostrar en pantalla “{TEMPC}℃ = {TEMPF}℉” Ir al paso 05.
5. Fin del algoritmo

Iteración	TEMPC	Salida
1	100	100°C = 212°F
2	0	0°C = 32°F
3	180	180°C = 356°F

Tabla. 03 – Prueba de escritorio del ejercicio 3. Nótese como en este caso el algoritmo acepta cualquier número real.

Logaritmos visuales

Recalcando lo que se había mencionado al principio de esta práctica, los logaritmos no sólo están escritos en pseudocódigo; estos también pueden usar lenguaje cotidiano, siempre y cuando este sea claro y conciso. También un algoritmo puede crear un resultado visual más amigable para los principiantes en la programación.

Por esto mismo se hicieron dos ejercicios de algoritmos para crear una estrella de cinco picos y un hexágono con ayuda de instrucciones en texto. Estos dos ejercicios se dibujaron de manera virtual y se adjuntan los videos para demostrar la manera en la que se arman ambos dibujos. Dichos enlaces están disponibles en GitHub.

Tarea

PROBLEMA: Encontrar los valores de X para una ecuación de segundo grado
RESTRICCIONES: Números reales.

DATOS DE ENTRADA: Tres números reales.
DATOS DE SALIDA: Las soluciones reales de la ecuación, si las hay.

DOMINIO: Todos los números reales.

1. Crear cinco variables; tres de ellas se llaman A, B & C. Asignarles los valores de A = 0, B = 0 & C = 0. Las dos variables restantes son X1 & X2. Ir al paso 02.
2. Mostrar en pantalla “Por favor introduzca los coeficientes de su ecuación. $Ax^2 + Bx + C = 0$”. Ir al paso 03.
3. Solicitar al usuario un número y almacenarlo como A. Ir al paso 04.
4. Solicitar al usuario un número y almacenarlo como B. Ir al paso 05.
5. Solicitar al usuario un número y almacenarlo como C. Ir al paso 06.

6. Obtener el valor de X1 a partir de la fórmula: X1 = (-B **+** SQRT(B - (4*A*C))) / 2A. Almacenar este valor numérico como X1. Ir al paso 07.
7. Obtener el valor de X2 a partir de la fórmula: X2 = (-B **-** SQRT(B - (4*A*C))) / 2A. Almacenar este valor numérico como X2. Ir al paso 08.
8. Si X1 = ERROR || X2 = ERROR mostrar en pantalla “La ecuación de segundo grado no tiene soluciones.” De lo contrario Ir al paso 09
9. Si X1 = X2 mostrar en pantalla “La solución única a esta ecuación es” {X1}. De lo contrario Ir al paso 10
10. Mostrar en pantalla “Las soluciones de esta ecuación son” {X1} “&” {X2}. Ir al paso 11.
11. Fin del algoritmo

Iteración	A	B	C	Salida
1	-1	10	-25	La solución única a esta ecuación es 5.
2	1	-2	-3	Las soluciones de esta ecuación son 3 & -1
3	2	12	30	La ecuación de segundo grado no tiene soluciones

Tabla. 04 – Prueba de escritorio del ejercicio de tarea. Nótese como el algoritmo puede lidiar con los tres casos al calcular las raíces de una ecuación de segundo grado.

GitHub

A continuación, la liga al repositorio de GitHub - https://github.com/CuellarJM/practica3_fdp/

Conclusión

Los algoritmos son la base de la programación, son las instrucciones que le dicen a una computadora cómo realizar una tarea. Aprender sobre algoritmos es crucial para la resolución de problemas, la eficiencia del código, el desarrollo del pensamiento computacional y la comprensión de los lenguajes de programación.

En conclusión, conocer estos fundamentos fortalece el pensamiento crítico y metódico, abre oportunidades laborales y proporciona una base sólida para el desarrollo de software.

Bibliografía

El lenguaje de programación C. Brian W. Kernighan, Dennis M. Ritchie, segunda edición, USA, Pearson Edu

Prabhu, R. (2023, August 3). What is algorithm: Introduction to algorithms. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-algorithms/

GitHub, Inc. (2024). Hello world. GitHub Docs. https://docs.github.com/en/get-started/start-your-journey/hello-world

Solano, J. A., García, E. E., & Montaño, L. S. (2022, February 21). Manual de prácticas del laboratorio de Fundamentos de programación. CDMX; Facultad de Ingeniería.