 



Actividad # 3| Código en lenguaje C

Introducción al desarrollo de Software

Ingeniería en Desarrollo de Software

TUTOR: Sandra Luz Lara Dévora

ALUMNO: Cristopher Eduardo Ramírez Calvillo

FECHA:08/08/2024

Índice

Introducción..................................................................................................................................... 3

Descripción....................................................................................................................................... 4

Justificación......................................................................................................................................5

Desarrollo.........................................................................................................................................6

Conclusión.........................................................................................................................................8

Referencias........................................................................................................................................9

Introducción

Crear calculadoras especificas en lenguaje C para números pares e impares, primos y números al revés implica una serie de procesos que combinan conocimientos matemáticos y habilidades de programación.

Para la calculadora de números primos, el objetivo fue identificar si el número es primo o no. Los números primos son aquellos que solo son divisibles entre 1 y por si mismos, el proceso generalmente incluye iterar desde 2 hasta la raíz cuadrada del número en cuestión, verificando si hay divisores. Si no se encuentran divisores, el número es primo, este método lo hace de manera eficiente y reduce el tiempo de cálculo.

En el caso de los números pares e impares, el proceso es más sencillo, lo cual, al momento de realizar tanto algoritmo, diagrama de flujo y codificación de lenguaje C requirió de menos campo y esfuerzo. El número es par únicamente si es divisible por 2, es decir, si el residuo de la división del número por 2 es 0. Si el residuo es diferente a 0, entonces el número es impar.

Por último, para la calculadora al revés, toma los números ingresados y lo invierte. Por ejemplo, al introducir el número 1234, la calculadora nos devolverá “el numero invertido es 4321”. El proceso implica convertir el número a una cadena de caracteres, invertir la cadena y luego convertirla nuevamente en números.

Descripción

Para la conversión de los diagramas de flujo a lenguaje C, seguimos una serie de pasos para asegurar que cada elemento del diagrama se haya traducido correctamente en instrucciones de programación. Veamos parte del proceso que llevamos a cabo:

Identificamos las variables que utilizamos en el diagrama, en este caso, cada uno de ellos hace una tarea en especifico independiente una de la otra (una define números primo, otra pares e impares y la ultima invierte el orden de los valores ingresados).

Usamos la función “main” que es el punto de entrada de cualquier programa en C, todo el código debe estar dentro de esta función, al igual que utilizamos la librería #include <stdio.h> para poder usar funciones de entrada y salida como “printf” y “scanf”.

En este caso, para representar decisiones utilizamos estructuras condicionales “if” aunque también teníamos a la mano el “switch” y en los ciclos usamos “for”, “while” o “do-while”.

1. P
2. OP
3. P

Justificación

Los diagramas de flujo permiten visualizar la lógica del programa de manera clara y estructurada. Al convertirlos a código, se asegura que esta lógica se implementa de manera precisa.

AL tener un diagrama de flujo como referencia, es menos probable cometer errores lógicos durante la codificación ya que actúa como guía paso a paso para los programadores, facilitando la traducción de ideas abstractas a instrucciones concretas en C.

Ayuda a descomponer problemas complejos en partes más manejables, lo que simplifica el proceso de codificación. Tener el diagrama de manera previa es sumamente importante para que se puedan localizar potenciales problemas o errores antes de llevarlo a cabo en lenguaje C.

La conversión de diagramas de flujo a código en C no solo asegura una implementación precisa y estructurada de la lógica del programa, sino que también mejora la comunicación, facilita la programación y optimiza el mantenimiento y la educación. Es un punto fundamental para transformar ideas abstractas en soluciones concretas y funcionales.

Desarrollo

Ahora llevaremos las actividades anteriores a codificación en lenguaje C como nos lo pide la actual:

Calculadora de números primos

Texto

Descripción generada automáticamente

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza bajaUna captura de pantalla de un celular

Descripción generada automáticamente con confianza media

Calculadora Par/Impar

Pantalla de computadora con letras

Descripción generada automáticamente con confianza mediaTexto

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Calculadora al revés

Texto

Descripción generada automáticamenteTexto

Descripción generada automáticamenteImagen que contiene naranja, alimentos, sostener, hombre

Descripción generada automáticamente

Conclusión

Pasar los diagramas de flujo a lenguaje C es una practica fundamental en la programación, ya que permite transformar una representación visual de un algoritmo a un código ejecutable. Esta conversión facilita la comprensión y el desarrollo de programas, asegurando que los pasos lógicos definidos en el diagrama se implementen correctamente en el código.

En la vida cotidiana, esta habilidad mejora la capacidad de resolver problemas de manera estructurada y eficiente. Al practicar la traducción de diagramas de flujo a código, se desarrolla un pensamiento lógico y analítico que es aplicable a diversas situaciones cotidianas, desde la planificación de tareas hasta la toma de decisiones complejas.

En cuanto a lo laboral, la capacidad de convertir diagramas de flujo en código C es altamente valorada. Los programadores que dominan esta habilidad pueden diseñar y desarrollar software de manera más eficiente, reduciendo errores y gastos, al igual que mejorando la calidad del producto final.

Referencias

Saavedra, J. A. (2023, June 1). *Qué es lenguaje C: las ventajas, las características y la sintaxis*. Ebac. https://ebac.mx/blog/que-es-lenguaje-c

Gálvez, J. a. S. (n.d.). *Fundamentos de Lenguaje C*. Unidades De Apoyo Para El Aprendizaje - CUAIEED - UNAM. https://uapa.cuaieed.unam.mx/sites/default/files/minisite/static/16a556ec-c557-42d1-a902-92e18cbdbf50/UAPA-fundamentos-lenguaje-c/index.html

Anexo de actividad en GitHub

*CristopherRamirez/codigo-en-Lenguaje-C*. (n.d.). https://github.com/CristopherRamirez/codigo-en-lenguaje-C