Universidad Corporativa Coppel | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad | 3 |

Nombre del curso

Ingeniería en Desarrollo de Software

TUTOR: Sandra Luz Lara Dévora

ALUMNO: Jesús Yahir Sojo Quiñonez

FECHA: 12/04/2025

Índice

Introducción

En esta actividad se realizará la codificación en lenguaje C de tres programas matemáticos previamente analizados: uno que identifica si un número es primo, otro que determina si un número es par o impar, y un tercero que invierte los dígitos de un número de 4 cifras. El objetivo principal es llevar a la práctica los conocimientos adquiridos sobre algoritmos, lógica de programación y estructuras de control, aplicándolos en un lenguaje real como lo es C. Esta actividad permite reforzar la importancia de la lógica secuencial y condicional en la resolución de problemas comunes. Asimismo, permite observar el funcionamiento de estructuras como ciclos for, condicionales if, y operadores básicos, los cuales son fundamentales para el desarrollo de programas funcionales. Esta práctica también fomenta la organización del código, la legibilidad y el análisis previo antes de la implementación.

Descripción

La actividad tiene como propósito la programación de cuatro ejercicios matemáticos utilizando el lenguaje C. Cada uno de estos programas tiene una funcionalidad específica basada en operaciones lógicas fundamentales. El primero permite determinar si un número es primo; el segundo identifica si un número ingresado es par o impar; y el tercero invierte los dígitos de un número de cuatro cifras. Esta actividad representa un avance en el proceso de aprendizaje, al trasladar la lógica construida en algoritmos y diagramas de flujo a un lenguaje de programación estructurado. Además, se pondrán en práctica estructuras como ciclos, condicionales y operadores matemáticos, que son esenciales en el desarrollo de cualquier tipo de software.

El desarrollo en C permite familiarizarse con un lenguaje de bajo nivel ampliamente utilizado en sistemas y aplicaciones. Con ello, se busca mejorar las habilidades técnicas del estudiante, fomentar el pensamiento lógico y prepararlo para resolver problemas reales de forma automatizada.

Justificación

El desarrollo de estos programas en lenguaje C no solo representa una práctica técnica, sino una herramienta pedagógica clave para afianzar conceptos fundamentales tanto en matemáticas como en programación. Al codificar funciones que determinan si un número es primo, par o impar, invertir cifras, se fortalecen habilidades como el análisis, la abstracción y la solución de problemas. Utilizar C como lenguaje base ofrece además una ventaja importante, ya que su sintaxis clara y su eficiencia lo convierten en una opción ideal para quienes están empezando a programar de forma estructurada.

Además, estas soluciones permiten visualizar de forma concreta cómo la lógica matemática puede automatizarse mediante programación, lo que abre posibilidades para crear herramientas más complejas en el futuro. Implementar este tipo de ejercicios ayuda a comprender cómo un problema se traduce en código, fortaleciendo las bases necesarias para el desarrollo de software más avanzado en la vida profesional.

Desarrollo – Codificación y ejecución

1. “Primos”

```c
#include <stdio.h>

int main() {
    int numero, i, contador, j;

    for(j = 1; j <= 10; j++) {
        contador = 0;
        printf("\n(%d) Ingresa un número: ", j);
        scanf("%d", &numero);

        for(i = 1; i <= numero; i++) {
            if(numero % i == 0) {
                contador++;
            }
        }

        if(contador == 2) {
            printf("Este número es primo\n");
        } else {
            printf("Este número no es primo\n");
        }
    }

    return 0;
}
```

```
(1) Ingresa un número: 1
Este número no es primo

(2) Ingresa un número: 2
Este número es primo

(3) Ingresa un número: 3
Este número es primo

(4) Ingresa un número: 4
Este número no es primo

(5) Ingresa un número: 5
Este número es primo

(6) Ingresa un número: 6
Este número no es primo

(7) Ingresa un número: 7
Este número es primo

(8) Ingresa un número: 8
Este número no es primo

(9) Ingresa un número: 9
Este número no es primo

(10) Ingresa un número: 10
Este número no es primo


=== Code Execution Successful ===
```

Explicación

El programa pide al usuario un número. Luego cuenta cuántos divisores tiene ese número. Si solo tiene dos (1 y él mismo), se considera primo. De lo contrario, no lo es.

2. **“Par/impar”**

```c
#include <stdio.h>

int main() {
    int numero, i;

    for(i = 1; i <= 10; i++) {
        printf("\n(%d) Ingresa un número: ", i);
        scanf("%d", &numero);

        if(numero % 2 == 0) {
            printf("Este número es par\n");
        } else {
            printf("Este número es impar\n");
        }
    }

    return 0;
}
```

```
(1) Ingresa un número: 1
Este número es impar

(2) Ingresa un número: 2
Este número es par

(3) Ingresa un número: 3
Este número es impar

(4) Ingresa un número: 4
Este número es par

(5) Ingresa un número: 5
Este número es impar

(6) Ingresa un número: 6
Este número es par

(7) Ingresa un número: 7
Este número es impar

(8) Ingresa un número: 8
Este número es par

(9) Ingresa un número: 9
Este número es impar

(10) Ingresa un número: 10
Este número es par


=== Code Execution Successful ===
```

Explicación

Este programa solicita un número al usuario y usa el operador % para saber si tiene residuo al dividirse entre 2. Si no tiene residuo, es par. Si tiene, es impar.

3. **“Al Revés”**

```c
#include <stdio.h>

int main() {
    int numero, d1, d2, d3, d4, invertido, i;

    for(i = 1; i <= 10; i++) {
        printf("\n(%d) Ingresa un número de 4 dígitos: ", i);
        scanf("%d", &numero);

        d1 = numero % 10;
        numero = numero / 10;

        d2 = numero % 10;
        numero = numero / 10;

        d3 = numero % 10;
        numero = numero / 10;

        d4 = numero;

        invertido = d1 * 1000 + d2 * 100 + d3 * 10 + d4;

        printf("El número al revés es: %d\n", invertido);
    }

    return 0;
}
```

```
(1) Ingresa un número de 4 dígitos: 1234
El número al revés es: 4321

(2) Ingresa un número de 4 dígitos: 2345
El número al revés es: 5432

(3) Ingresa un número de 4 dígitos: 3456
El número al revés es: 6543

(4) Ingresa un número de 4 dígitos: 4567
El número al revés es: 7654

(5) Ingresa un número de 4 dígitos: 5678
El número al revés es: 8765

(6) Ingresa un número de 4 dígitos: 5487
El número al revés es: 7845

(7) Ingresa un número de 4 dígitos: 6598
El número al revés es: 8956

(8) Ingresa un número de 4 dígitos: 2365
El número al revés es: 5632

(9) Ingresa un número de 4 dígitos: 4521
El número al revés es: 1254

(10) Ingresa un número de 4 dígitos: 1597
El número al revés es: 7951


=== Code Execution Successful ===
```

Explicación

El programa descompone el número ingresado en sus cifras y las reorganiza para mostrarlas al revés. Se usa división y módulo (%) para extraer cada dígito.

Conclusión

La realización de esta actividad me permitió comprender de manera práctica cómo aplicar la lógica de programación para resolver problemas matemáticos básicos mediante el lenguaje C. A través del desarrollo de los programas para identificar números primos, clasificar números como pares o impares, invertir números de cuatro cifras y calcular factoriales, pude reforzar el uso de estructuras de control como ciclos for, condiciones if, y operadores aritméticos.

Además, esta práctica resulta muy útil tanto en el ámbito académico como en el profesional, ya que estas operaciones forman parte de tareas más complejas en áreas como el análisis de datos, desarrollo de software educativo o incluso en pruebas técnicas para entrevistas de programación.

El hecho de repetir cada programa 10 veces me ayudó a validar el funcionamiento de los algoritmos con diferentes entradas, y esto es algo que también se valora en entornos reales de trabajo donde la verificación y validación del código es fundamental. En conclusión, esta actividad fue una excelente manera de reforzar mis conocimientos de programación en C, desarrollar mi lógica algorítmica y preparar mejor mis habilidades para futuros retos en el campo del desarrollo de software.

https://github.com/YahirSojo/Introducci-n-al-Desarrollo-de-Software.git