

Actividad | #3 | Código en Lenguaje

C Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Devora

ALUMNO: Juan Antonio Valenzuela Castillo

FECHA: 20/04/2025

Índice

Pagina

1.-Introduccion.....	3
2.-Descripcion.....	4
3.-Justificacion.....	5
4.-Desarrollo.....	6
○ Codificación	
○ Ejecución en el compilador	
5.-Conclusion.....	9

Introducción

En esta actividad se desarrollarán tres programas matemáticos utilizando el lenguaje de programación C. Cada uno de estos programas está diseñado con un objetivo específico: identificar si un número es primo, clasificar diez números como pares o impares, e invertir un número de cuatro dígitos.

Estos ejercicios permiten aplicar estructuras de control básicas como condicionales y ciclos, fundamentales en la programación. Asimismo, fortalecen la lógica matemática y la capacidad de resolución de problemas de forma automatizada. La finalidad es comprender cómo se pueden implementar conceptos numéricos mediante algoritmos eficientes y claros, a través del uso de herramientas computacionales.

Este tipo de prácticas también permite al estudiante familiarizarse con la estructura y sintaxis del lenguaje C, que es uno de los lenguajes base en el estudio de las ciencias computacionales. Esta experiencia práctica representa un paso importante para el desarrollo de habilidades en programación estructurada.

Descripción

La actividad presentada busca fortalecer el razonamiento lógico y el pensamiento algorítmico mediante la programación de tres soluciones matemáticas en lenguaje C. Cada uno de los programas tiene una función particular: el primero determina si un número ingresado por el usuario es primo, el segundo clasifica diez números como pares o impares, y el tercero invierte un número de cuatro cifras.

Estas tareas, aunque sencillas en apariencia, representan un reto importante para la aplicación correcta de estructuras de control como condicionales, ciclos y manejo de variables. Además, permiten comprender cómo se traduce un problema del mundo real al lenguaje computacional, desarrollando habilidades clave para cualquier estudiante de ingeniería o carreras relacionadas con tecnología.

Esta actividad no solo sirve como práctica de codificación, sino que también entrena la capacidad de analizar problemas, proponer soluciones lógicas, y estructurar procesos paso a paso. Es un ejercicio ideal para consolidar los fundamentos de la programación estructurada.

Justificación

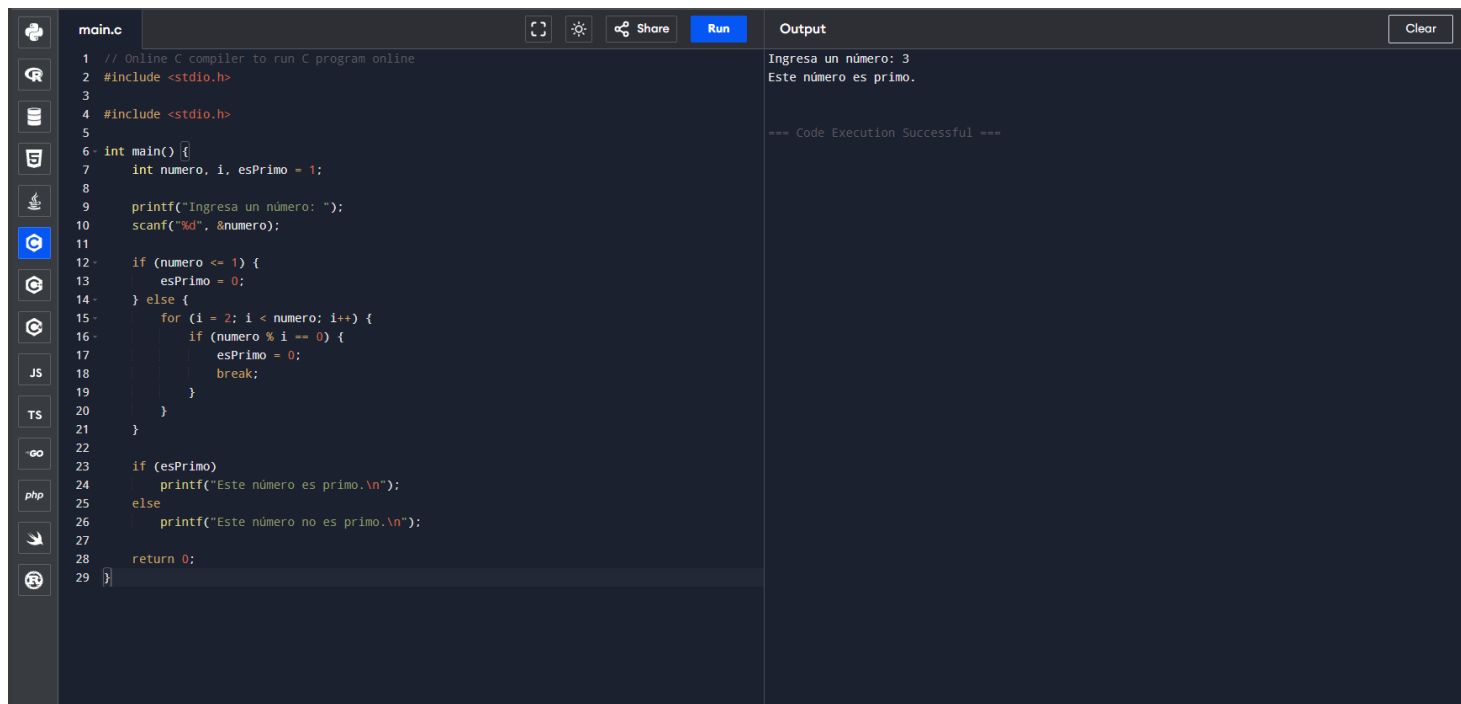
El desarrollo de estos programas matemáticos en lenguaje C representa una estrategia efectiva para introducir al estudiante en la lógica de programación y el pensamiento computacional. La elección de problemas como identificar números primos, clasificar pares e impares e invertir cifras no es arbitraria: se trata de situaciones matemáticas comunes que pueden resolverse mediante algoritmos sencillos pero potentes.

Estos ejercicios permiten al estudiante familiarizarse con las estructuras básicas de programación, como condicionales (if, else), ciclos (for, while), y operaciones aritméticas. Además, trabajar con el lenguaje C refuerza una comprensión profunda de cómo funciona el software a nivel más cercano al hardware, ya que es un lenguaje de bajo nivel en comparación con otros más modernos.

Implementar estas soluciones no solo ayuda a resolver problemas matemáticos con mayor eficiencia, sino que también sienta las bases para resolver problemas más complejos en el futuro. Por tanto, esta actividad se justifica como una herramienta pedagógica fundamental para el aprendizaje de la programación estructurada.

Desarrollo

La siguiente imagen se mostrará el lenguaje de programación C, de la calculadora de números “primos”



```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 #include <stdio.h>
5
6 int main() {
7     int numero, i, esPrimo = 1;
8
9     printf("Ingresa un número: ");
10    scanf("%d", &numero);
11
12    if (numero <= 1) {
13        esPrimo = 0;
14    } else {
15        for (i = 2; i < numero; i++) {
16            if (numero % i == 0) {
17                esPrimo = 0;
18                break;
19            }
20        }
21    }
22
23    if (esPrimo)
24        printf("Este número es primo.\n");
25    else
26        printf("Este número no es primo.\n");
27
28    return 0;
29 }
  
```

Output

```

Ingresa un número: 3
Este número es primo.

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

Ejecución. –

En la imagen se muestra la ejecución del programa que identifica si un número es primo. El usuario ingresa un número entero cuando el programa lo solicita con el mensaje "**Ingresa un número**". Luego, el programa evalúa si el número es divisible solo entre 1 y él mismo, y muestra el mensaje correspondiente. Si el número es primo, aparece en pantalla "**El número X sí es primo**". En caso contrario, muestra "**El número X no es primo**". En la captura se debe ver claramente el número ingresado, el resultado y el código fuente.

La siguiente imagen se mostrará el lenguaje de programación C, de la calculadora de “par/impar”

The screenshot shows an online C compiler interface. On the left, there is a sidebar with icons for various programming languages (C, JS, TS, PHP, etc.). The main area is divided into two panels: 'main.c' on the left and 'Output' on the right. The 'main.c' panel contains the following C code:

```
1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 #include <stdio.h>
5
6 int main() {
7     int numero, i;
8
9     for (i = 1; i <= 10; i++) {
10        printf("Ingresa el número %d: ", i);
11        scanf("%d", &numero);
12
13        if (numero % 2 == 0)
14            printf("Este número es par.\n");
15        else
16            printf("Este número es impar.\n");
17    }
18
19    return 0;
20 }
```

The 'Output' panel shows the results of the program's execution:

```
Ingresa el número 1: 2
Este número es par.
Ingresa el número 2: 3
Este número es impar.
Ingresa el número 3: 12
Este número es par.
Ingresa el número 4: 79
Este número es impar.
Ingresa el número 5:
```

Ejecución. –

En esta ejecución se observa un ciclo que se repite 10 veces, ya que el programa solicita ingresar 10 números. Cada vez que se introduce un número, el programa evalúa si es par o impar utilizando el operador módulo. Si el resultado es cero, muestra “**Este número es par**”, y si no lo es, muestra “**Este número es impar**”. En la captura se deben mostrar al menos 3 o 4 ejemplos de números ingresados y sus respectivas respuestas para ilustrar cómo se ejecuta el ciclo y se imprime el resultado en pantalla.

La siguiente imagen muestra el lenguaje de programación C, de la calculadora “numero invertidos”

```

1 // Online C compiler to run C program online
2 #include <stdio.h>
3
4 #include <stdio.h>
5
6 int main() {
7     int numero, invertido = 0;
8
9     printf("Ingresa un número de 4 dígitos: ");
10    scanf("%d", &numero);
11
12    if (numero < 1000 || numero > 9999) {
13        printf("El número no tiene 4 dígitos.\n");
14        return 1;
15    }
16
17    while (numero != 0) {
18        invertido = invertido * 10 + numero % 10;
19        numero /= 10;
20    }
21
22    printf("El número al revés es: %d\n", invertido);
23
24    return 0;
25 }
  
```

Output

```

Ingresa un número de 4 dígitos: 1234
El número al revés es: 4321

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

Ejecución. –

En esta captura se muestra la ejecución del programa que invierte un número de cuatro cifras. El usuario ingresa un número dentro del rango de 1000 a 9999, y el programa utiliza una serie de divisiones y residuos para reorganizar sus cifras al revés. El mensaje en pantalla debe ser: **“El número al revés es: XXXX”**, mostrando el número correctamente invertido. La imagen debe incluir el número original ingresado, el resultado final, y parte del código fuente para que quede documentado cómo funciona el programa.

Conclusión

Al finalizar esta actividad, puedo decir que trabajar con el lenguaje de programación C fue una experiencia muy útil para seguir desarrollando mis habilidades en lógica y resolución de problemas. Aunque al principio se me hizo un poco complicado entender algunos conceptos, como la sintaxis exacta o cómo estructurar los programas, con práctica y dedicación logré sacar adelante cada uno de los ejercicios propuestos.

Me pareció interesante ver cómo, con unas pocas líneas de código, se pueden resolver problemas matemáticos de forma rápida y precisa. Además, me ayudó a entender mejor cómo funciona un programa desde cero, desde el análisis del problema hasta su codificación y pruebas.

Esta actividad me sirvió no solo para aplicar lo que he aprendido en clase, sino también para darme cuenta de lo importante que es tener paciencia y practicar constantemente. En resumen, fue una experiencia muy provechosa que me motivó a seguir aprendiendo más sobre programación.