

Actividad | #3 | – Códigos en Lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Lara Devora

ALUMNO: Francisco Antonio Herrera Silvas

FECHA: 30/11/2024

DESCRIPCION 4

JUSTIFICACION 5

DESARROLLO 6

 NUMEROS PRIMOS 6

 PAR E IMPAR 7

 NUMEROS INVERTIDOS 8

CONCLUSION..... 9

INTRODUCCION

En la presente actividad se desarrollarán y analizarán diversos códigos escritos en lenguaje de programación C, diseñados para resolver problemas específicos y facilitar la comprensión de conceptos fundamentales de programación. El lenguaje C, conocido por su eficiencia, flexibilidad y cercanía al hardware, es ampliamente utilizado en el desarrollo de aplicaciones que requieren alto rendimiento, como sistemas operativos, aplicaciones embebidas y herramientas de software de bajo nivel. Este ejercicio tiene como propósito reforzar habilidades técnicas, fomentar el pensamiento lógico y proporcionar ejemplos prácticos de cómo aplicar la programación para resolver problemas cotidianos y técnicos.

Los códigos presentados en esta actividad están enfocados en tres funcionalidades principales:

1. Cálculo de números primos: Este código permitirá identificar si un número ingresado por el usuario es primo, utilizando conceptos básicos de divisibilidad y control de flujo. El propósito es demostrar cómo se puede verificar propiedades matemáticas a través de algoritmos simples.
2. Determinación de paridad: Aquí se programará un sistema que evalúe una lista de 10 números ingresados por el usuario, clasificándolos en pares o impares. Este ejercicio ilustra el uso de estructuras iterativas y condicionales para trabajar con conjuntos de datos.
3. Inversión de números: Este código invertirá un número entero de cuatro dígitos ingresado por el usuario. Este proceso implica descomponer el número en sus componentes, reorganizarlos y reconstruir el resultado, proporcionando una demostración clara de cómo manipular datos numéricos y trabajar con operaciones matemáticas básicas.

A lo largo de esta actividad, se explicará la lógica detrás de cada código, destacando las mejores prácticas en programación, como la claridad en la escritura del código, el manejo de entradas y salidas, y la optimización de recursos computacionales. La finalidad es no solo resolver problemas específicos, sino también construir una base sólida para abordar desafíos más complejos en futuros proyectos de programación.

DESCRIPCION

La actividad planteada se centra en el desarrollo de programas utilizando el lenguaje de programación C, con el objetivo de abordar problemas específicos y reforzar habilidades fundamentales de programación. Este enfoque permite comprender cómo las herramientas computacionales pueden ser empleadas para resolver desafíos cotidianos y técnicos. El contexto propuesto incluye tres programas que representan distintas problemáticas: identificar números primos, determinar la paridad de una lista de números y realizar la inversión de un número de cuatro dígitos.

El diseño de estas aplicaciones refleja una metodología didáctica orientada al aprendizaje práctico. Cada tarea implica la implementación de algoritmos que no solo cumplen con un propósito funcional, sino que también promueven el desarrollo de habilidades como el pensamiento lógico, el manejo de estructuras de control y la manipulación de datos. Estas competencias son esenciales para cualquier programador y constituyen la base para abordar problemas más complejos en el futuro.

Por ejemplo, el código que identifica números primos permite explorar la lógica matemática detrás de los divisores y su implementación en un programa, mientras que la determinación de paridad de números demuestra la importancia de las estructuras de control iterativas y condicionales. Finalmente, el programa que invierte números introduce conceptos como la descomposición y reconstrucción de datos, fundamentales para trabajar con datos numéricos y sus transformaciones.

Lo solicitado en esta actividad no solo busca la resolución técnica de los problemas, sino también una comprensión profunda del proceso lógico y la capacidad para justificar cada paso en el desarrollo del código. Esto fomenta el pensamiento crítico y la autovaloración del aprendizaje, al permitir que el estudiante evalúe sus propios avances y decisiones.

JUSTIFICACION

El empleo de soluciones basadas en el lenguaje de programación C para la actividad propuesta se justifica por varios motivos, tanto técnicos como pedagógicos. Este enfoque permite resolver problemas específicos de manera eficiente y, al mismo tiempo, fomenta el desarrollo de competencias esenciales en el ámbito de la programación.

En primer lugar, el lenguaje C es ampliamente reconocido por su simplicidad, potencia y versatilidad, lo que lo convierte en una herramienta ideal para enseñar los fundamentos de la programación. Su sintaxis clara y estructurada facilita el aprendizaje de conceptos clave como estructuras de control, manejo de datos y diseño de algoritmos. Estos conceptos son indispensables para abordar las problemáticas presentadas, como la identificación de números primos, la determinación de paridad y la inversión de un número.

Además, C permite trabajar a un nivel más cercano al hardware, lo que ofrece a los estudiantes una visión clara de cómo se gestionan los recursos computacionales, como la memoria y el procesamiento. Esto resulta particularmente útil en actividades como esta, donde es importante optimizar el rendimiento del código y garantizar su correcta ejecución en diversos entornos.

Desde un enfoque pedagógico, desarrollar estas soluciones en C fomenta el pensamiento lógico y estructurado. Los estudiantes deben analizar cada problema, descomponerlo en pasos manejables y traducir esos pasos en un lenguaje que la computadora pueda entender. Este proceso no solo les ayuda a resolver los ejercicios planteados, sino que también los prepara para enfrentar desafíos más complejos en el futuro.

Finalmente, las soluciones implementadas en C son portables y pueden ejecutarse en una amplia gama de sistemas operativos y dispositivos. Esto asegura que los programas desarrollados en esta actividad puedan ser fácilmente utilizados o adaptados para otros contextos, maximizando su utilidad práctica.

DESARROLLO

NUMEROS PRIMOS

CODIGO EN C

```
#include<stdio.h>

int main() {
    int contador;
    int i;
    int num;
    /* definir variables */
    contador = 0;
    /* pedir al usuario que ingrese un numero */
    printf("ingrese un numero para verificar si es primo\n");
    scanf("%i", &num);
    /* verificar si el numero es primo */
    for (i=1; i<=num; ++i) {
        if ((num%i==0)) {
            contador = contador+1;
        }
    }
    /* Un número primo solo es divisible por 1 y por sí mismo */
    if (contador==2) {
        printf("El número %i sí es primo.\n", num);
    } else {
        printf("El número %i no es primo.\n", num);
    }
    return 0;
}
```

Capturas del codigo en ejecucion

```
ingrese un numero para verificar si es primo
3
El número 3 sí es primo.
```

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

```
ingrese un numero para verificar si es primo
10
El número 10 no es primo.
```

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

PAR E IMPAR

```
#include<stdio.h>

int main() {
    int i;
    int impares;
    int numero;
    int pares;
    pares = 0;
    impares = 0;
    /* Bucle para ingresar 10 números */
    for (i=1; i<=10; ++i) {
        printf("Ingresa el número %i:\n", i);
        scanf("%i", &numero);
        /* Verificar si el número es par o impar */
        if ((numero%2==0)) {
            printf("El número %i es par.\n", numero);
            pares = pares+1;
        } else {
            printf("El número %i es impar.\n", numero);
            impares = impares+1;
        }
    }
    /* Mostrar la cantidad de números pares e impares */
    printf("De los 10 números ingresados:\n");
    printf("Números pares: %i\n", pares);
    printf("Números impares: %i\n", impares);
    return 0;
}
```

```
Ingresa el número 1:
1
El número 1 es impar.
Ingresa el número 2:
2
El número 2 es par.
Ingresa el número 3:
3
El número 3 es impar.
Ingresa el número 4:
4
El número 4 es par.
Ingresa el número 5:
5
El número 5 es impar.
Ingresa el número 6:
6
El número 6 es par.
Ingresa el número 7:
7
El número 7 es impar.
Ingresa el número 8:
8
El número 8 es par.
Ingresa el número 9:
9
El número 9 es impar.
```

NUMEROS INVERTIDOS

```
#include <stdio.h>

int main() {
    int num, reversed = 0, remainder;

    printf("Ingrese un número de 4 dígitos: ");
    scanf("%d", &num);

    if (num < 1000 || num > 9999) {
        printf("Por favor, ingrese un número de 4 dígitos.\n");
        return 1;
    }

    while (num != 0) {
        remainder = num % 10;
        reversed = reversed * 10 + remainder;
        num /= 10;
    }

    printf("Número invertido: %d\n", reversed);
    return 0;
}
```

```
Ingrese un número de 4 dígitos: 4321
Número invertido: 1234
```

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


CONCLUSION

La actividad realizada, enfocada en el desarrollo de programas en lenguaje C para resolver problemas específicos, tiene una importancia significativa tanto en el ámbito laboral como en la vida cotidiana. A través de este ejercicio, se fortalecen habilidades prácticas de programación y se promueve un pensamiento lógico y estructurado, elementos esenciales en el campo profesional de las tecnologías de la información y en la resolución de problemas cotidianos.

En el contexto laboral, las competencias adquiridas mediante este tipo de actividades son altamente valoradas. La capacidad de diseñar, implementar y depurar algoritmos eficientes es fundamental para roles como desarrollador de software, ingeniero de sistemas o especialista en datos. Los problemas abordados en esta actividad —como identificar números primos, determinar la paridad de valores y manipular datos numéricos— representan conceptos básicos que forman la base para resolver desafíos más complejos en áreas como optimización de sistemas, diseño de bases de datos y análisis de datos. Estas habilidades permiten a los profesionales construir soluciones innovadoras, mejorar procesos existentes y adaptarse a las demandas cambiantes del mercado tecnológico.

En la vida cotidiana, el desarrollo de soluciones algorítmicas fomenta el pensamiento crítico y la creatividad para resolver problemas. Por ejemplo, la capacidad de identificar patrones, analizar datos y encontrar soluciones optimizadas es útil en la planificación de proyectos personales, gestión financiera o incluso en la toma de decisiones informadas en actividades diarias. Este enfoque metódico y lógico puede aplicarse más allá del ámbito tecnológico, convirtiéndose en una herramienta valiosa para abordar desafíos en diversas áreas de la vida.

Además, trabajar con un lenguaje como C permite a los estudiantes y profesionales comprender cómo funcionan los sistemas computacionales a nivel básico. Esto no solo les brinda un entendimiento profundo de las tecnologías que utilizamos diariamente, sino que también les capacita para construir soluciones más eficientes y adaptadas a necesidades específicas.

Finalmente, esta actividad destaca la importancia de aprender a programar como una habilidad transversal y adaptable. En un mundo cada vez más digitalizado, la capacidad de resolver problemas computacionales no solo es relevante para el sector tecnológico, sino que también abre puertas a oportunidades en campos como la educación, la medicina, la industria automotriz y las ciencias, entre otros.