



Programas
Académicos



Actividad 1 - Algoritmos

Introducción al desarrollo de software

Ingeniería en Desarrollo de Software.

Tutor: Felipe de Jesús Araux López

Alumno:

Luis José Monroy Rodríguez G2 Luisjo2000mr@hotmail.com


Fecha: 27/07/2023

Índice

Introducción	2
Descripción	3
Justificación.....	4
Desarrollo	5
Primos	5
Par/impar	6
Al Revés	7
Conclusión.....	9
Referencias	10

Introducción

Un algoritmo es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades, dado un estado inicial y una entrada, siguiendo los pasos sucesivos se llega a un estado final y se obtiene una solución, por lo tanto, la presente actividad funcionará en base a este tema, los algoritmos se caracterizan por tener un inicio y un fin, funcionar en secuencia, ser preciso y no ambiguo, ser abstracto y tener una cantidad finita de pasos, tomando en cuenta lo anterior mencionado partiremos con una actividad que trata de la creación de los algoritmos básicos de tres calculadoras cuyos procesos son distintos y su explicación minuciosa, con el fin de que el espectador sienta la comodidad de que tendrá un aprendizaje de una manera sencilla y precisa de cómo se harán los algoritmos de dichos ejemplares.



Descripción

En la presente actividad se realizará el primer algoritmo de una calculadora que se llamará Primos, su objetivo será identificar los números primos que se ingresen para esto se tomará en cuenta la lógica matemática adecuada para poder reflejarlo en el algoritmo de una manera entendible y aplicable para el espectador y el algoritmo deberá imprimir “El número si es primo” en caso de que no, imprimirá “El número no es primo”, el segundo algoritmo será para una calculadora llamada Par/Impar,, su objetivo es que se ingresen 10 números, ya sean pares o impares y deberá imprimir “es par” o “es impar” dependiendo el número, el tercer algoritmo será para un programa llamado Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los número al revés o invertidos. Por ejemplo, si se ingresa el número 4568, el programa matemático deberá regresar 8654. de igual forma que los anteriores algoritmos, se implementará y se explicará la lógica matemática para que el espectador comprenda cómo se llegó a la conclusión de este proceso para que pueda dar el resultado.

Justificación

La tarea de programar tres tipos de calculadoras diferentes, llamadas Primos, Par/Impar y Al Revés, es una solución efectiva para ayudar a los usuarios a aprender y practicar conceptos matemáticos básicos. Estas calculadoras están diseñadas para identificar números primos, determinar si un número es par o impar y revertir un número de 4 dígitos, respectivamente. Al utilizar algoritmos para resolver estos problemas matemáticos, las calculadoras pueden proporcionar soluciones rápidas y confiables. Además, el uso de algoritmos asegura que las calculadoras sean precisas, eficientes y fáciles de usar para los usuarios. En resumen, el empleo de algoritmos para programar estas calculadoras es una solución efectiva para ayudar a los usuarios a aprender y practicar conceptos matemáticos básicos., de igual forma, la gente profesionalista tiene grandes ventajas si al tener ya dominado el tema, los beneficios y desarrollos son exponenciales ya que este tipo de aprendizaje son el presente y el futuro de los proyectos de distintas empresas que tengan como objetivo avanzar.

Desarrollo

Primos

Inicio

- 1.- Leer el número ingresado por el usuario
- 2.- Inicializar una variable “cont” en 0
- 3.- Para cada número entero i desde 1 hasta el número ingresado:
 - 3.1.- Si el número ingresado es divisible por i, incrementar “cont” en 1
- 4.- Si “cont” es mayor a 2, imprimir “El número no es primo”
- 5.- Si no, imprimir “El número es primo”

Fin

Este algoritmo determina si un número ingresado por el usuario es primo o no.

Primero, el algoritmo lee el número ingresado por el usuario y luego inicializa una variable “cont” en 0.

Luego, el algoritmo utiliza un bucle “for” para iterar sobre cada número entero “i” desde 1 hasta el número ingresado.

Dentro del bucle, si el número ingresado es divisible por “i”, la variable “cont” se incrementa en 1.

Después de que se completa el bucle, el algoritmo verifica si “cont” es mayor a 2. Si es así, el algoritmo imprime que el número no es primo.

De lo contrario, imprime que el número es primo.

La lógica detrás de este algoritmo se basa en la definición de un número primo: un número entero mayor que 1 que solo es divisible por 1 y por sí mismo.

Al contar la cantidad de divisores del número ingresado y verificar si es mayor a 2, el algoritmo puede determinar si el número es primo o no.

Este algoritmo verifica si un número es primo comprobando si tiene más de dos divisores (1 y el mismo). Si encuentra más de dos divisores, el algoritmo concluye que el número no es primo y lo imprime.

Si no encuentra más de dos divisores, el algoritmo concluye que el número es primo y lo imprime.

Par/impar

Inicio

1.- Inicializar una lista vacía para almacenar los 10 números.

2.- Para cada número de 1 a 10:

2.1.- Pedir al usuario que ingrese un número entero.

2.2.- Agregar el número a la lista.

3.- Para cada número en la lista:

3.1.- Si el número es divisible por 2, imprimir que es un número par.

3.2.- Si no, imprimir que es un número impar.

Fin

Este algoritmo determina si cada uno de los 10 números ingresados por el usuario es par o impar.

Primero, el algoritmo inicializa una lista vacía para almacenar los 10 números.

Luego, el algoritmo utiliza un bucle “for” para iterar sobre cada número de 1 a 10.

Dentro del bucle, el algoritmo pide al usuario que ingrese un número entero y luego agrega ese número a la lista.

Después de que se completa el bucle, el algoritmo utiliza otro bucle “for” para iterar sobre cada número en la lista.

Dentro de este bucle, si el número es divisible por 2, el algoritmo imprime que es un número par.

De lo contrario, imprime que es un número impar. La lógica detrás de este algoritmo se basa en la definición de un número par: un número entero que es divisible por 2.

Al verificar si cada número en la lista es divisible por 2, el algoritmo puede determinar si cada número es par o impar.

Al Revés

Inicio

- 1.- Pedir al usuario que ingrese un número entero de 4 dígitos.
- 2.- Inicializar una variable invertido en 0.
- 3.- Mientras el número ingresado sea mayor que 0:
 - 3.1.- Multiplicar invertido por 10.
 - 3.2.- Sumar a invertido el último dígito del número ingresado.
 - 3.3.- Eliminar el último dígito del número ingresado dividiéndolo entre 10.
- 4.- Imprimir el valor de invertido.

Fin

Este algoritmo invierte un número entero de 4 dígitos ingresado por el usuario.

Primero, el algoritmo pide al usuario que ingrese un número entero de 4 dígitos y luego inicializa una variable “invertido” en 0.

Luego, el algoritmo utiliza un bucle “while” para iterar mientras el número ingresado sea mayor que 0.

Dentro del bucle, el algoritmo multiplica “invertido” por 10 y luego suma a “invertido” el último dígito del número ingresado.

Después, el algoritmo elimina el último dígito del número ingresado dividiéndolo entre 10.

Después de que se completa el bucle, el algoritmo imprime el valor de “invertido”.

La lógica detrás de este algoritmo se basa en la manipulación de los dígitos del número ingresado para invertir su orden.

Al multiplicar “invertido” por 10 y sumarle el último dígito del número ingresado en cada iteración del bucle, el algoritmo construye el número invertido dígito por dígito.

Como dato adicional, aquí un ejemplo escrito:

- 1.- Supongamos que el usuario ingresa el número entero de 4 dígitos 1234.
- 2.- Inicializa una variable “invertido” en 0. “invertido”
- 3.- Mientras el número ingresado sea mayor que 0:
 - 3.1.- Multiplica “invertido” por 10: $\text{“invertido”} = 0 * 10 = 0$.
 - 3.2.- Suma el último dígito del número ingresado a “invertido”: $\text{“invertido”} = 0 + 4 = 4$.
 - 3.3.- Elimina el último dígito del número ingresado dividiéndolo entre 10: $1234 // 10 = 123$.
- 4.- Repite el paso 3 hasta que el número ingresado ya no sea mayor que 0.
- 5.- El valor final de “invertido” es 4321, que es el reverso del número ingresado.

El bucle “for” es útil para realizar tareas repetitivas, como iterar sobre elementos en una matriz o generar una secuencia de números y el bucle “while” es útil para realizar tareas repetitivas cuyo número de repeticiones no se conoce de antemano, como leer datos de un archivo hasta llegar al final o esperar a que se cumpla una condición específica.

Conclusión

La actividad de programar tres tipos de calculadoras diferentes, llamadas Primos, Par/Impar y Al Revés, es importante tanto en el campo laboral como en la vida cotidiana. En el campo laboral, estas calculadoras pueden ser utilizadas por profesionales en áreas como la educación, la ingeniería y las finanzas para ayudar a los estudiantes y colegas a aprender y practicar conceptos matemáticos básicos y para también poder hacer lucro significativo en base a poder enseñar estos conocimientos, en la vida cotidiana, estas calculadoras pueden ser utilizadas por cualquier persona para mejorar su comprensión de los conceptos matemáticos y para resolver problemas matemáticos de manera rápida y precisa. En resumen, la realización de esta actividad es importante porque proporciona una herramienta útil para ayudar a las personas a aprender y practicar conceptos matemáticos básicos en su campo laboral y en su vida cotidiana y así poder tener ese As bajo la manga que la mayoría necesita y tiene que empezar a implementar.

Referencias

- Wikipedia (s.f.). Algoritmo. Recuperado el 27 de julio de 2023, de <https://es.wikipedia.org/wiki/Algoritmo>
- Significados.com (s.f.). Algoritmo. Recuperado el 27 de julio de 2023, de <https://www.significados.com/algoritmo/>
- Programa Resuelto. (2022, 4 de mayo). #17 [C] – Leer un número de cuatro cifras e imprimirlo al revés [Vídeo]. YouTube.
https://www.youtube.com/watch?v=aAIGNZJ2iE&ab_channel=ProgramaResuelto