

Unidad 1 – Recursividad

La resolución de problemas debe hacerse en lenguaje C++

Ejercicio N°1

Implemente una función para potencias enteras con recursividad. Definición recursiva para elevar un número a una potencia: Un número elevado a la potencia cero produce la unidad; la potencia de un número se obtiene multiplicándolo por sí mismo elevando a la potencia menos uno. Por ejemplo:

$$3^2 = 3 * (3^1) = 3 * [3 * (3^0)] = 3 * (3 * 1) = 9$$

Ejercicio N°2

Implementar una función recursiva que determine la suma de los “n” primeros números naturales ej: para n=5 mostrar por pantalla 1+2+3+4+5 y luego el resultado de la suma.

Ejercicio N°3

Implementar una función recursiva que dado un número entero lo muestre de forma invertida ej: 10523 de mostrar 32501. Dar aviso de error en caso de uso de números negativos y mostrar primero el número a invertir y después el número invertido.

Ejercicio N°4

Crear un programa que emplee recursividad para calcular el mayor de los elementos de un vector de “n” tamaño dado por pantalla por el usuario.

Ejercicio N°5

Crear un programa que pida al usuario un número entero positivo “n” y un dígito “d”. Luego crear una función recursiva que determine si “d” está en “n”. Si está, la función debe retornar verdadero y si no está debe retornar falso.

Ej.: si n=1323 y d = 5, retorna Falso y si d= 1 retorna verdadero.

Ejercicio N°6

Crear un programa que pida al usuario un número entero positivo “n”. Luego implementar una función recursiva que cuente la cantidad de dígitos pares que ocupan posiciones impares (identificándolas de izquierda a derecha) en “n”. Ej.: si el número es 22005 el resultado es 2, y si fuera 1414 el resultado es 0

Ejercicio N°7: Búsqueda de una Clase en el Horario Escolar.

Eres un estudiante que necesita encontrar el horario de una clase específica en tu horario semanal de clases. Estás interesado en comparar dos tipos de búsquedas, la secuencial y la binaria. Usando recursividad, implementa el código en lenguaje C++. Se puede usar la librería String.

Búsqueda Secuencial Recursiva:

- Implementa una función recursiva que busque la clase en una lista (Vector y Librería String) desordenada de clases y horarios y cuente las comparaciones (if) realizadas.

Búsqueda Binaria Recursiva:

- Implementa una función recursiva que busque la clase en una lista (Vector y Librería String) ordenada de clases y horarios y cuente las comparaciones (if) realizadas.

Requisitos

- El programa debe solicitar al usuario ingresar el número de clases, las clases y los horarios en la lista (Vector y Librería String). Considerar que se necesitarán un vector para las clases y otro vector para los horarios.
- El programa debe imprimir el nombre de la clase encontrada y el horario que le corresponde. También debe imprimir el número de comparaciones (if) realizadas en cada tipo de búsqueda.
- El programa debe impedir que se ingrese una lista desordenada para la búsqueda binaria. Pedile al usuario que ingrese en orden alfabético y que el programa verifique que así sea.

Escribe un programa en C++ que realice lo siguiente:

Búsqueda Secuencial Recursiva:

- Implementa una función recursiva que busque un número en un arreglo de enteros utilizando la búsqueda secuencial.
- Cuenta el número de comparaciones (if) realizadas durante la búsqueda.

Búsqueda Binaria Recursiva:

- Implementa una función recursiva que busque un número en un arreglo de enteros ordenados utilizando la búsqueda binaria.
- Cuenta el número de comparaciones (if) realizadas durante la búsqueda.

Requisitos

- El programa debe solicitar al usuario ingresar el tamaño del arreglo, los elementos del arreglo, y el número a buscar.
- El programa debe imprimir el índice en el que se encuentra el número buscado (o -1 si no se encuentra) y el número de comparaciones (if) realizadas en cada tipo de búsqueda.

Ejercicio N°8:

Ejercicio: Procesamiento de Tareas Diarias

Imagina que tienes una lista de tareas diarias. Cada elemento de esa lista tiene el nombre de una tarea diaria. Un estudio determinó que el esfuerzo necesario para una tarea puede estimarse con la cantidad de caracteres con la que se nombra. Por lo cual, deseas programar un código en C++ que te determine cuál es la tarea del listado que te implica mayor esfuerzo para ejecutarla primero.

Para ello, implementarás un código usando un enfoque iterativo y otro código con enfoque recursivo.

Iterativo: Implementa una función que utilice un bucle para contar el número total de caracteres en una lista de tareas.

Recursivo: Implementa una función recursiva que realice la misma tarea.

En ambos casos, mida la cantidad de llamadas recursivas y de iteraciones e imprimirlas en pantalla para comparar los dos métodos.