**Linked List Program Documentation**

**Overview**

This C++ program implements a singly linked list structure with functionality to:

1. Insert nodes from a file, user input, or randomly generated numbers.
2. Sort the linked list using the merge sort algorithm.
3. Print the list and determine its size.

The program begins by prompting the user to choose an option for populating the linked list, then performs sorting and displays both the original and sorted lists.

**Components**

**Headers and Namespaces**

A screen shot of a computer

Description automatically generated

This section includes necessary libraries:

* **iostream** for input/output
* **fstream** for file operations
* **time.h** for random number generation
* **vector** for storing list items temporarily Aliasing cout and cin simplifies syntax, eliminating the need for std:: prefix.

**Data Structure Definition**

**A black screen with white text

Description automatically generated**

**This defines a node struct, representing each node in the linked list. Each node contains:**

* **data: Stores an integer value.**
* **next: Pointer to the next node in the list.**

**Function Prototypes**

**These functions manage different aspects of the linked list:**

1. **RandomList: Generates a list with random integers.**
2. **FileInsert: Reads integers from a file and inserts them into the list.**
3. **MergeSort: Sorts the linked list using merge sort.**
4. **Merge: Merges two sorted lists.**
5. **Split: Splits a list into two halves for merge sort.**
6. **Insert: Inserts a new node into the list.**
7. **Print: Prints all nodes in the list.**
8. **sizeOf: Returns the size of the list.**
9. **UserInput: Captures user input for list data.**

**Main Function**

**A screen shot of a computer program

Description automatically generated**

**A black rectangular object with a black strip

Description automatically generated**

**Function Details**

**UserInput**

**Prompts the user for an integer and returns it.**

**FileInsert**

**Reads a list of integers from a specified file and inserts them into the list.**

* **Parameters:**
  + **path: File path for reading data.**
  + **vectorList: Pointer to the linked list.**
* **Returns: Updated linked list with file data.**

**RandomList**

**Generates a linked list with random integers.**

* **Returns: Linked list populated with random values between 1 and 100.**

**MergeSort**

**Sorts the linked list using merge sort by recursively splitting and merging.**

* **Parameters: Pointer to the linked list.**

**Merge**

**Merges two sorted lists.**

* **Parameters:**
  + **leftList: Left sublist.**
  + **rightList: Right sublist.**
* **Returns: Sorted merged list.**

**Split**

**Splits the list into two halves by using two pointers, fast and slow.**

* **Parameters:**
  + **source: Source list.**
  + **frontRef: First half of the list.**
  + **backRef: Second half of the list.**

**Insert**

**Inserts a node at the start of the list.**

* **Parameters:**
  + **pos: Pointer to the list.**
  + **info: Value to insert.**

**Print**

**Prints each node’s data.**

* **Parameters: Pointer to the list.**

**sizeOf**

**Returns the size (number of nodes) in the linked list.**

* **Parameters: Pointer to the list**