

In this document I will describe the functionality of my code and what each function does.

In this project I created 3 files: Algorithms\_Individual#1.cpp, Source.cpp, Header.h.

Header.h:

This part contains the structure definition, type alias, and function prototypes used in the program.  
Declares prototypes for functions implemented in "Source.cpp" and "Algorithms\_Individual#1."

Source.cpp:

This file contains the function implementations defined in "Header.h."

Algorithms\_Individual#1.cpp:

This file contains the main function and serves as the entry point of the program.

I will show the input and output of functions in my code.

```
Enter the number of elements in the linked list: 5
Enter the value of element 1: 1
Enter the value of element 2: 2
Enter the value of element 3: 3
Enter the value of element 4: 4
Enter the value of element 5: 5
Enter the number of elements in the array: 4
Enter the values of array elements:
1
2
3
5
Number of connected components in the array: 2
```

ListNode struct defines a basic element for a singly linked list. It has an integer value and a pointer to the next node in the list. When a new node is created, it's initialized with a value and its next pointer is initially set to nullptr.

- 1) createLinkedList() - function that is responsible for creating a singly linked list
- 2) contains(const vector<int>& vals, int target) - function checks whether a specific integer target is present in a given vector of integers (vals).
- 3) countConnectedComponents(ListNode\* head, vector<int>& vals) - function is responsible for counting the number of connected components in a singly linked list, based on whether the values in the linked list nodes are present in a given vector. And the contains() function is used as a helper function within the countConnectedComponents(ListNode\* head, vector<int>& vals) function to determine whether the value of each node in the linked list is present in the vals vector.

4) `main()`:

In the main function, the program:

- Calls `createLinkedList()` to create a linked list based on user input.
- Prompts the user to enter the number of elements in an array and the array elements.
- Calls `countConnectedComponents(linkedList, arrayElements)` to determine the number of connected components in the linked list based on the array values.
- Outputs the number of connected components in the array.
- Cleans up memory.