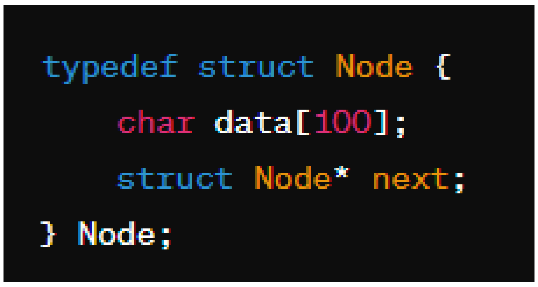
Script

Intro: Assalamuyalaikum everyone. (Introduce yourself and your team member).
Today I am going to present our project called **“Simple Search Engine”**. It was made by Data structure called **“Linked List”.** I'm here with [Person B] and [Person C]. Together, we'll walk you through the code, explain how it works, and demonstrate a simple search functionality within our linked list."

Person B: Code Explanation Part 1

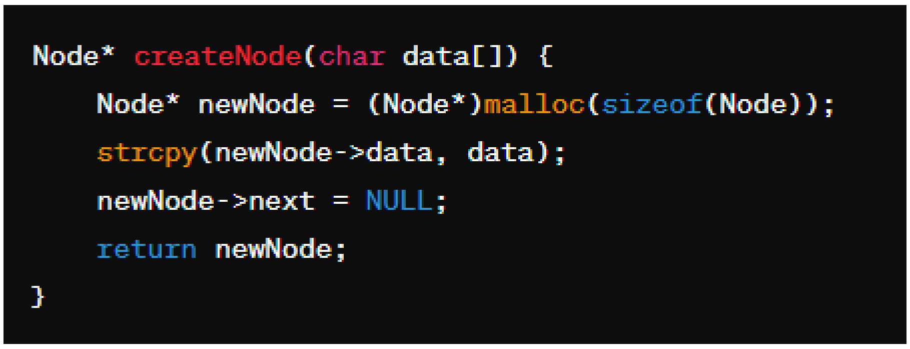
Person B:
"Thank you, [Person A]. Let's start by looking at the core structure of our linked list. In our code, we define a structure for each node of the linked list:



```c
typedef struct Node {
    char data[100];
    struct Node* next;
} Node;
```

Here, **data** is an array of characters to store **strings**, and next is a **pointer** to the **next node** in the list.

Next, we have a function to create a new node:

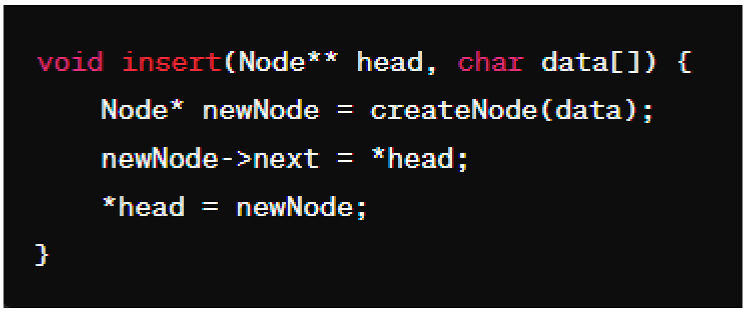


```c
Node* createNode(char data[]) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    strcpy(newNode->data, data);
    newNode->next = NULL;
    return newNode;
}
```

This function allocates memory for a new node, copies the input string into the node's data by the function called **strcpy()**, and initializes the next pointer to NULL."

Person C: Code Explanation Part 2

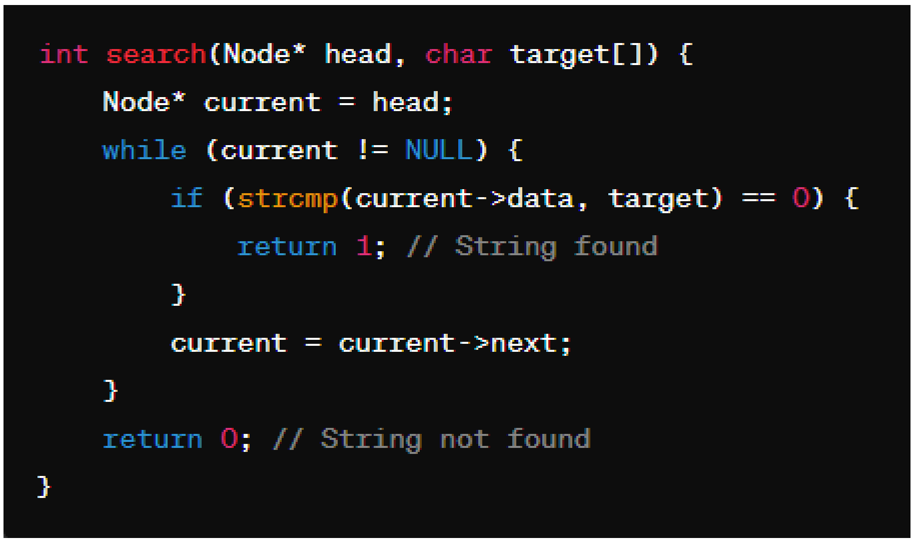
Person C:
"Great, [Person B]. Now let's see how we insert nodes into our linked list. The insert function adds a new node at the beginning of the list:



```c
void insert(Node** head, char data[]) {
    Node* newNode = createNode(data);
    newNode->next = *head;
    *head = newNode;
}
```

In this function, we create a new node with the given data, set its next pointer to the current head of the list, and then update the head to point to this new node.

To search for a string in the linked list, we use the search function:



```c
int search(Node* head, char target[]) {
    Node* current = head;
    while (current != NULL) {
        if (strcmp(current->data, target) == 0) {
            return 1; // String found
        }
        current = current->next;
    }
    return 0; // String not found
}
```

This function traverses the list, comparing each node's data with the target string. If a match is found, it returns 1; otherwise, it returns 0 after checking all nodes."

Person A: Demonstration and Conclusion

Person A:
"Thank you, [Person B] and [Person C]. Now, let's see how this all comes together in the main function:

```c
int main() {
    Node* head = NULL;

    // Populate the linked list with some sample strings
    insert(&head, "apple");
    insert(&head, "banana");
    insert(&head, "orange");
    insert(&head, "grape");
    insert(&head, "kiwi");

    // Ask the user to enter a string to search for
    char target[100];
    printf("Enter a string to search for: ");
    scanf("%s", target);

    // Perform the search
    if (search(head, target)) {
        printf("String '%s' found in the list.\n", target);
    } else {
        printf("String '%s' not found in the list.\n", target);
    }

    return 0;
}
```

In main, we first create an empty linked list and then populate it with some sample strings using the insert function. We prompt the user to enter a string, and then we search for that string in the list using the search function. The result of the search is printed to the console.

That's a brief overview of our project. Thank you for listening, and we're happy to answer any questions you might have."

Person B and Person C: "Thank you!"