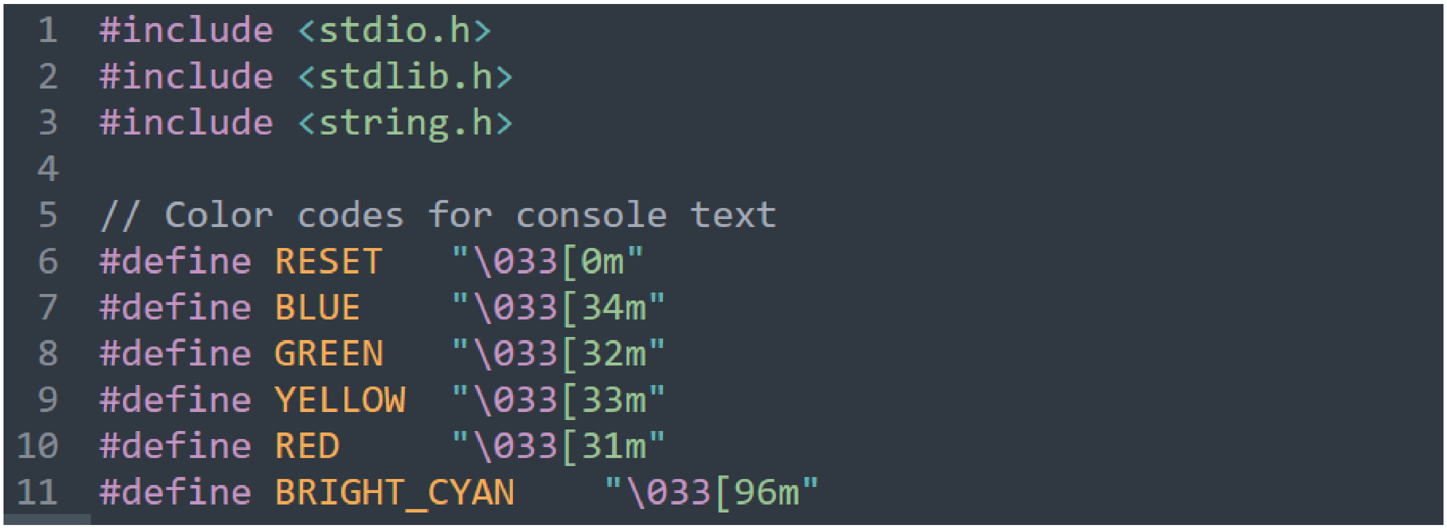
Linked list assignment

1. Including Libraries and Defining Colors



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Color codes for console text
#define RESET   "\033[0m"
#define BLUE    "\033[34m"
#define GREEN   "\033[32m"
#define YELLOW  "\033[33m"
#define RED     "\033[31m"
#define BRIGHT_CYAN    "\033[96m"
```

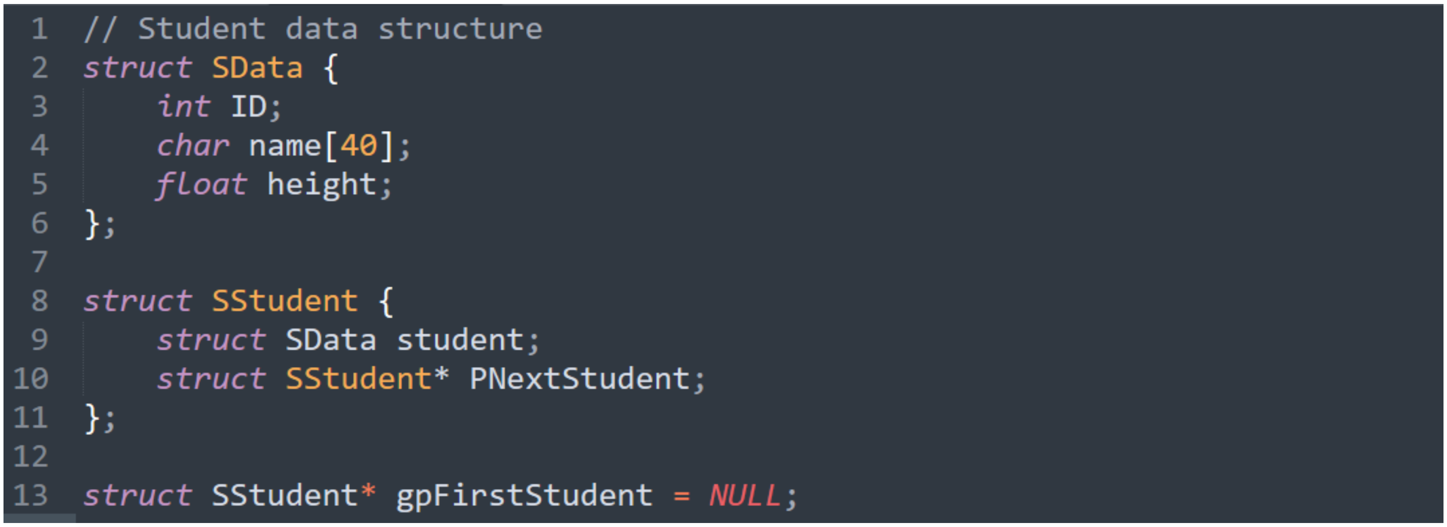
Libraries:

- #include <stdio.h>: Includes standard input and output functions.
- #include <stdlib.h>: Includes functions for memory allocation and other system-related functions.
- #include <string.h>: Includes functions for handling strings.

Color Codes:

- Defines color codes for text in the terminal to enhance readability and visual appeal.

2. Defining Data Structures



```c
// Student data structure
struct SData {
    int ID;
    char name[40];
    float height;
};

struct SStudent {
    struct SData student;
    struct SStudent* PNextStudent;
};

struct SStudent* gpFirstStudent = NULL;
```

SData Structure:

- Contains the student's data, including ID, name, and height.

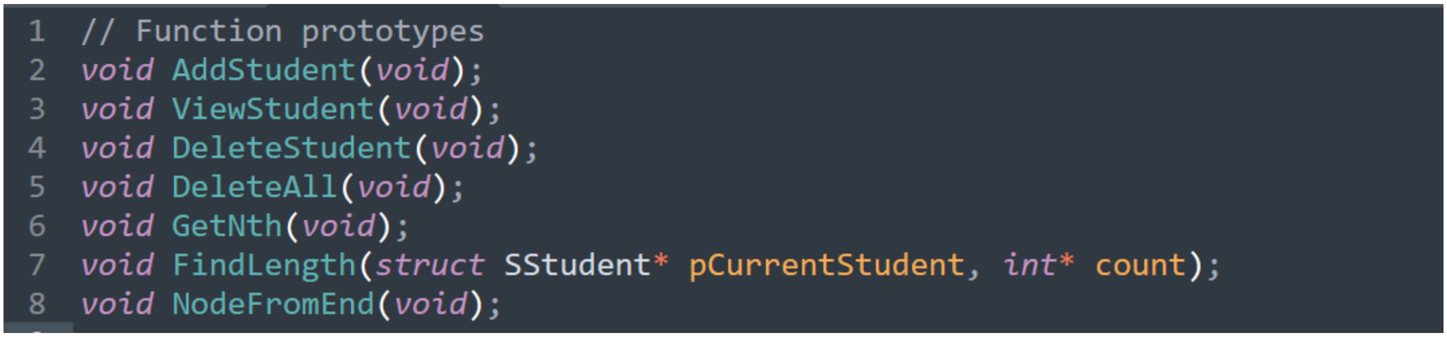
SStudent Structure:

- Contains the SData structure for student details and a pointer to the next student in the linked list.

gpFirstStudent Pointer:

- A pointer to the first student in the list, initialized to NULL because the list starts empty.

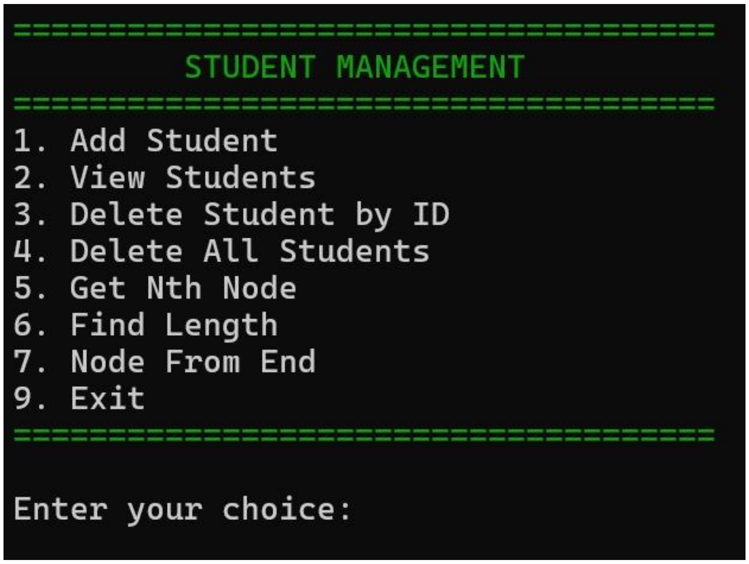
3. Function Prototypes



```c
// Function prototypes
void AddStudent(void);
void ViewStudent(void);
void DeleteStudent(void);
void DeleteAll(void);
void GetNth(void);
void FindLength(struct SStudent* pCurrentStudent, int* count);
void NodeFromEnd(void);
```

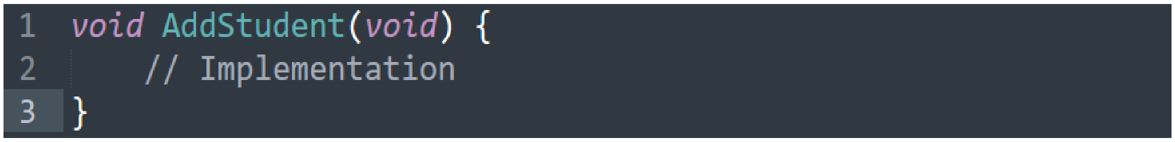
Function Prototypes:

- Declarations of functions that handle adding, viewing, deleting, and retrieving students in the linked list.



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice:
```

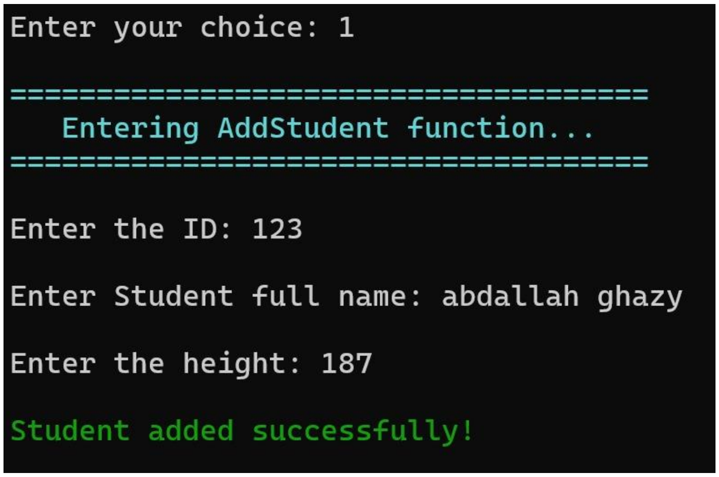
4. Function to Add a Student



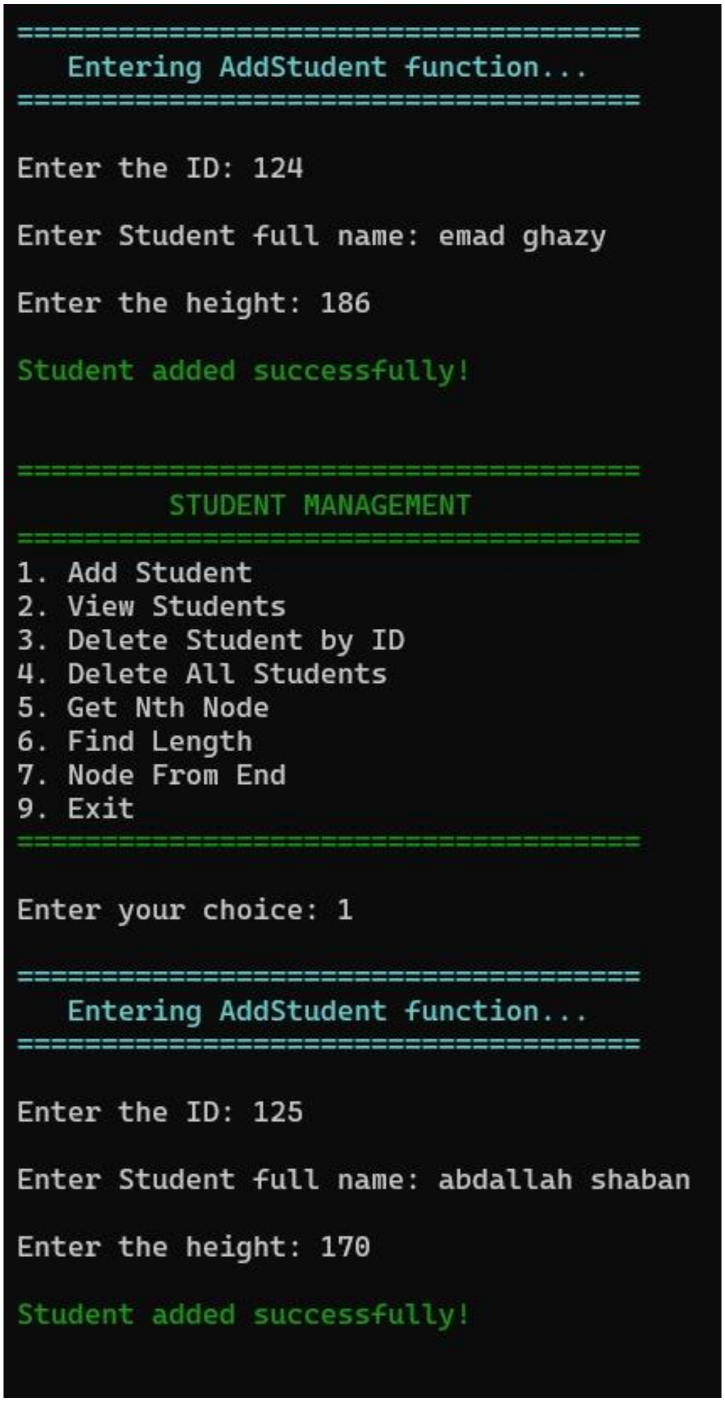
```c
void AddStudent(void) {
    // Implementation
}
```

Explanation:

- This function adds a new student to the end of the linked list. It checks memory allocation, collects student data from the user, and appends the student to the list.

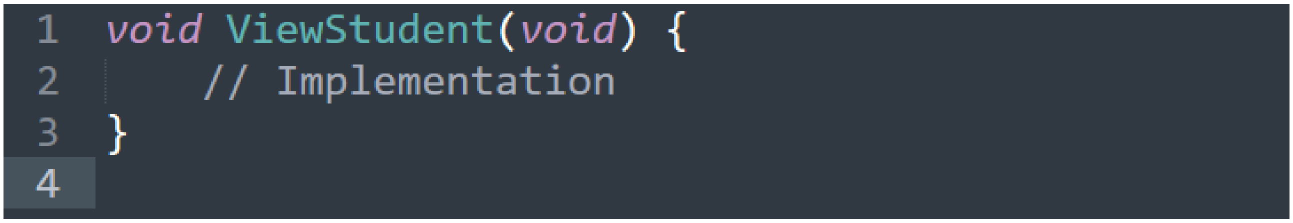


```
Enter your choice: 1

=====================================
   Entering AddStudent function...
=====================================

Enter the ID: 123

Enter Student full name: abdallah ghazy

Enter the height: 187

Student added successfully!
```



```
=====================================
   Entering AddStudent function...
=====================================

Enter the ID: 124

Enter Student full name: emad ghazy

Enter the height: 186

Student added successfully!


=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 1

=====================================
   Entering AddStudent function...
=====================================

Enter the ID: 125

Enter Student full name: abdallah shaban

Enter the height: 170

Student added successfully!
```

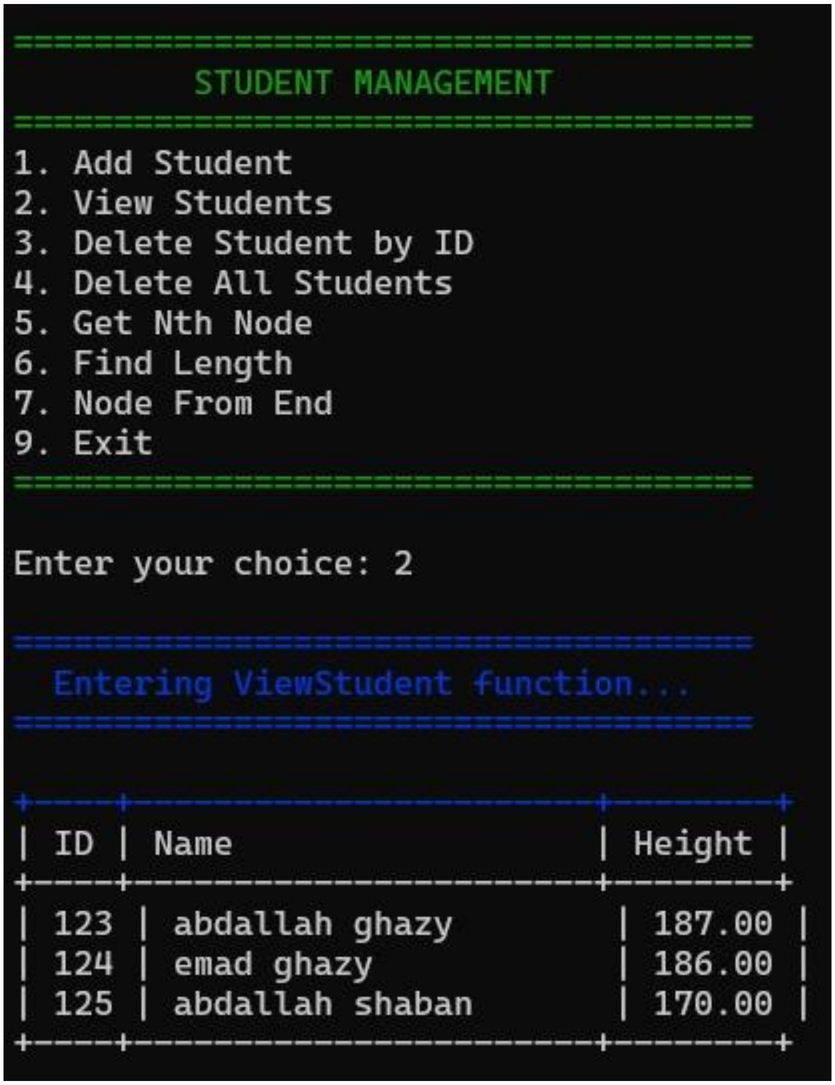
5. Function to View All Students



```c
void ViewStudent(void) {
    // Implementation
}
```

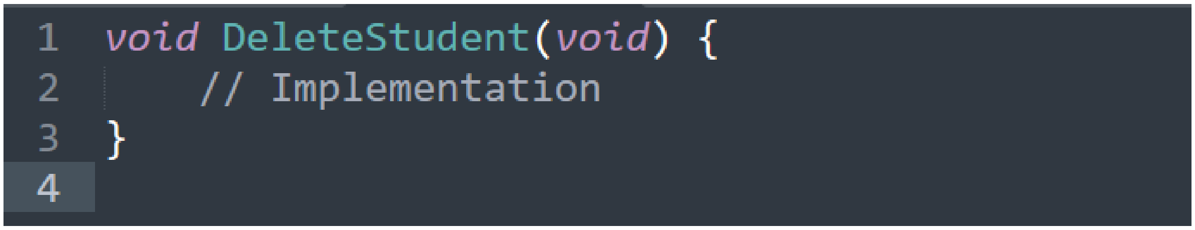
Explanation:

- This function displays all students in the linked list. It prints a table of student data if the list is not empty.



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 2

=====================================
  Entering ViewStudent function...
=====================================

+----+----------------------+--------+
| ID | Name                 | Height |
+----+----------------------+--------+
| 123 | abdallah ghazy       | 187.00 |
| 124 | emad ghazy           | 186.00 |
| 125 | abdallah shaban      | 170.00 |
+----+----------------------+--------+
```

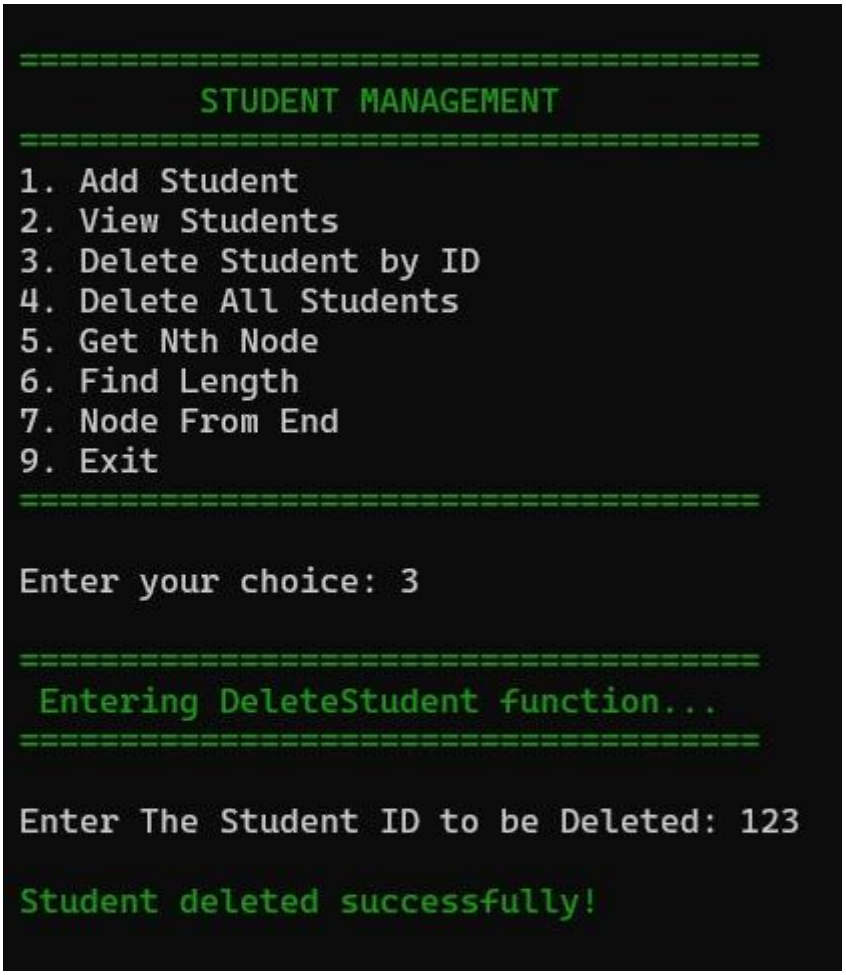
6. Function to Delete a Student by ID



```c
void DeleteStudent(void) {
    // Implementation
}
```

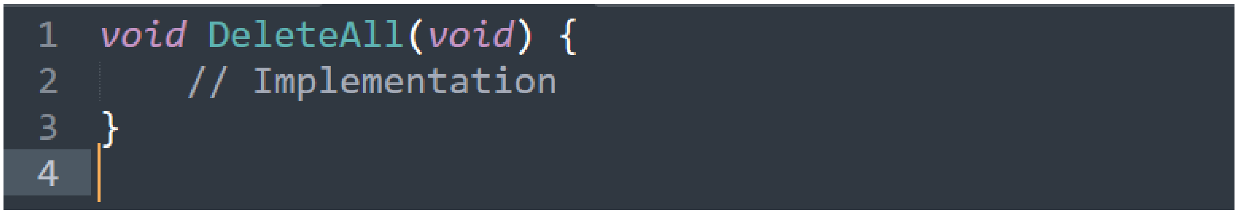
Explanation:

- This function removes a student from the list based on the provided student ID. It updates pointers accordingly and frees the memory allocated for the student



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 3

=====================================
 Entering DeleteStudent function...
=====================================

Enter The Student ID to be Deleted: 123

Student deleted successfully!
```

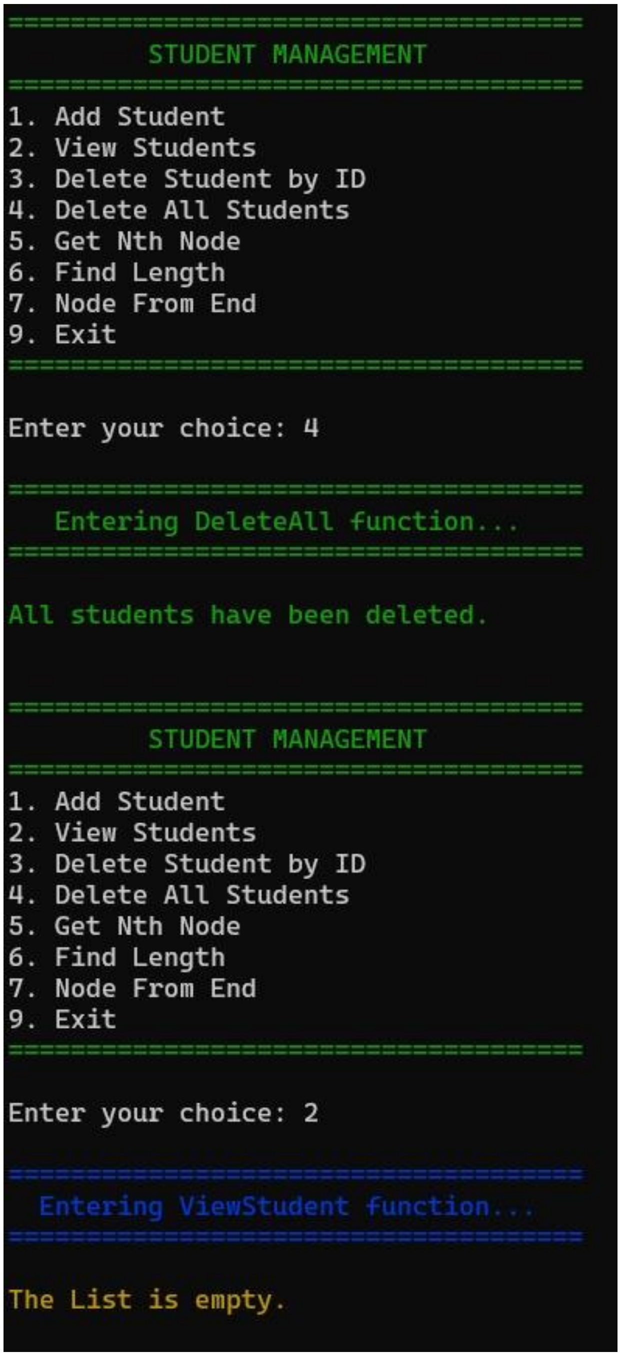
7. Function to Delete All Students



```c
void DeleteAll(void) {
    // Implementation
}
```

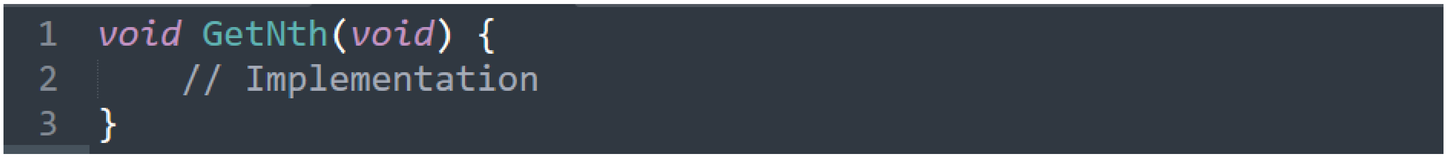
Explanation:

- This function deletes all students from the list and frees the memory allocated for each student, effectively clearing the list.



```
=====================================
          STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 4

=====================================
   Entering DeleteAll function...
=====================================

All students have been deleted.


=====================================
          STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 2

=====================================
  Entering ViewStudent function...
=====================================

The List is empty.
```

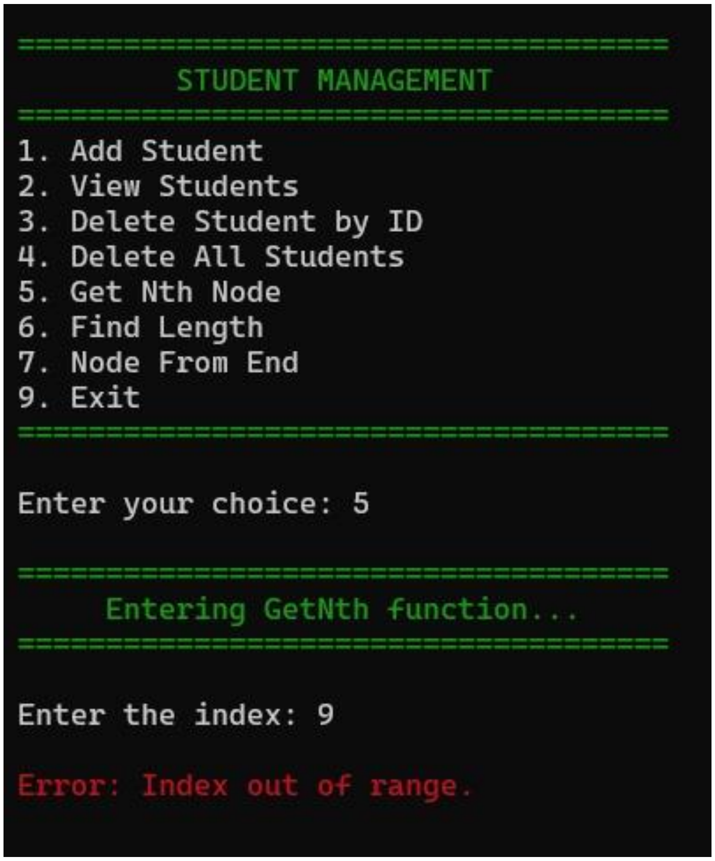
8. Function to Get the Nth Student



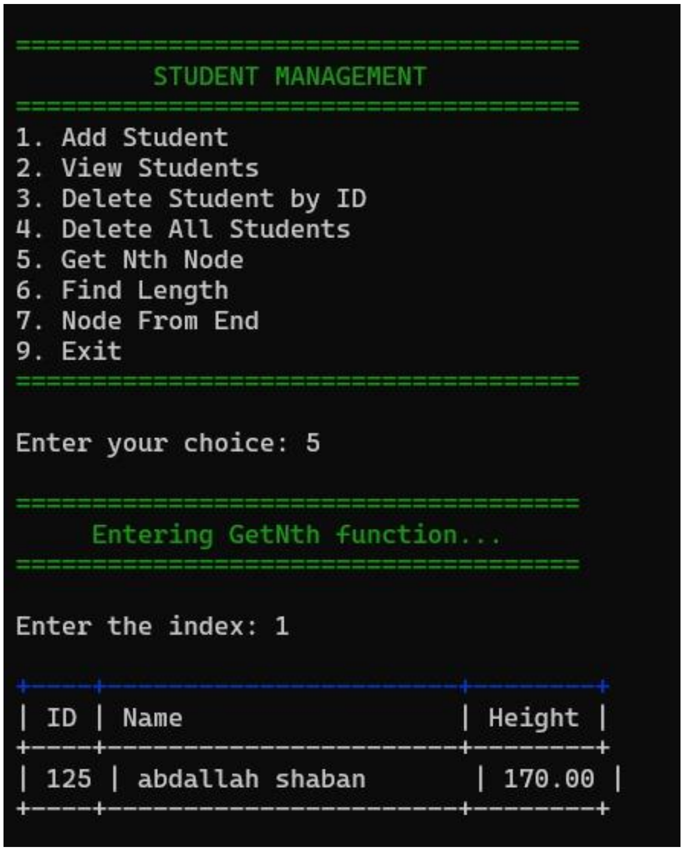
```c
void GetNth(void) {
    // Implementation
}
```

Explanation:

- This function retrieves and displays the student at the specified index from the list. It ensures the index is valid and the list is not empty.

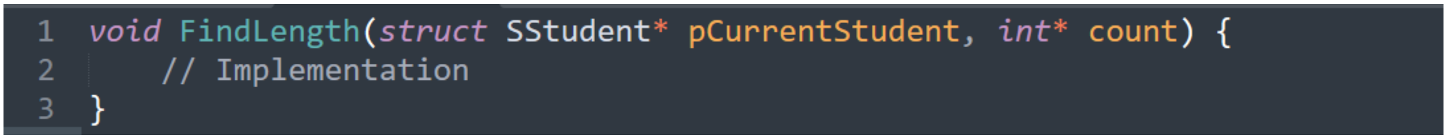


```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 5

=====================================
     Entering GetNth function...
=====================================

Enter the index: 9

Error: Index out of range.
```



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 5

=====================================
     Entering GetNth function...
=====================================

Enter the index: 1

+-----+-----------------------+---------+
| ID  | Name                  | Height  |
+-----+-----------------------+---------+
| 125 | abdallah shaban       | 170.00 |
+-----+-----------------------+---------+
```

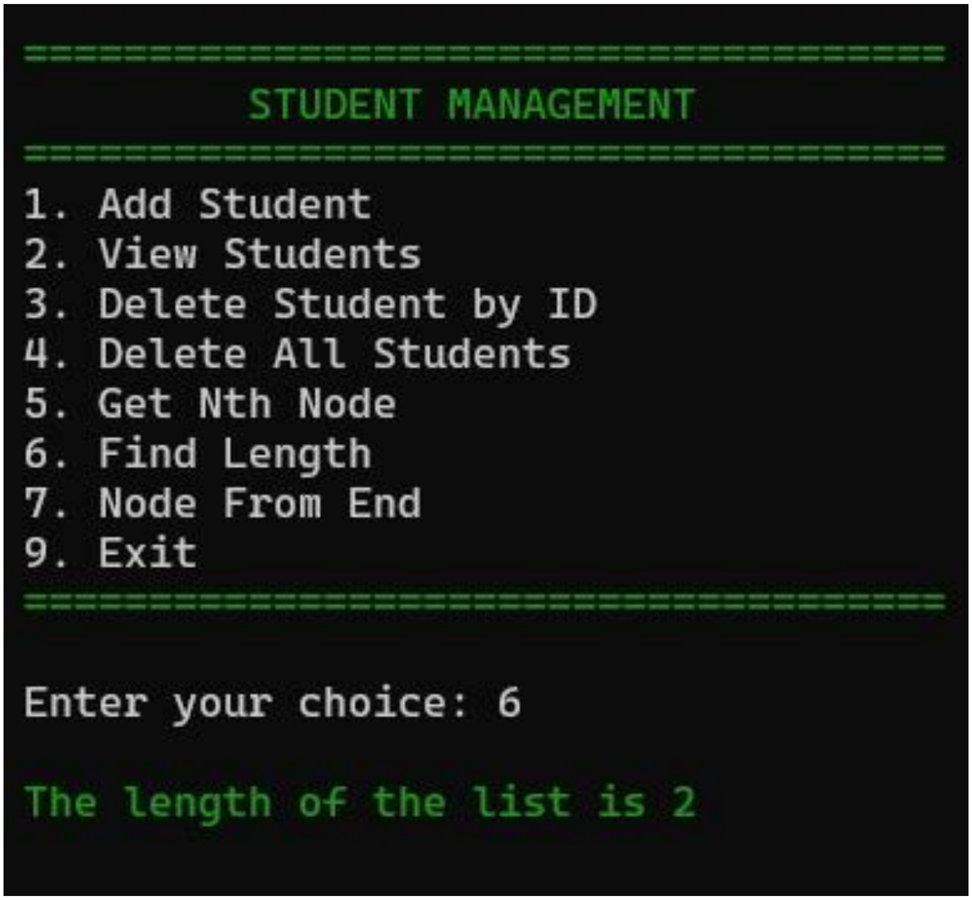
9. Function to Find the Length of the List



```c
void FindLength(struct SStudent* pCurrentStudent, int* count) {
    // Implementation
}
```

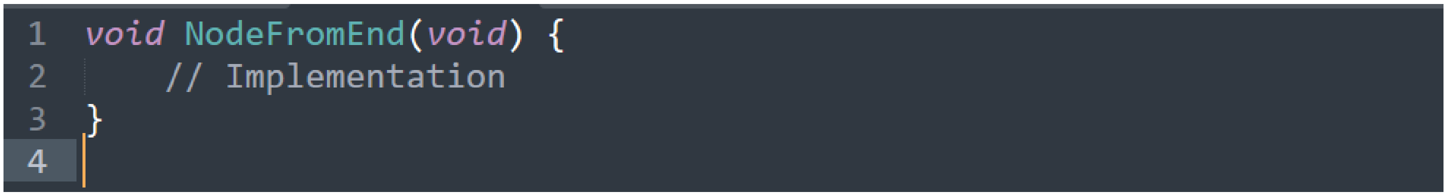
Explanation:

- This recursive function calculates the length of the list by traversing it and counting the number of students.



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 6

The length of the list is 2
```

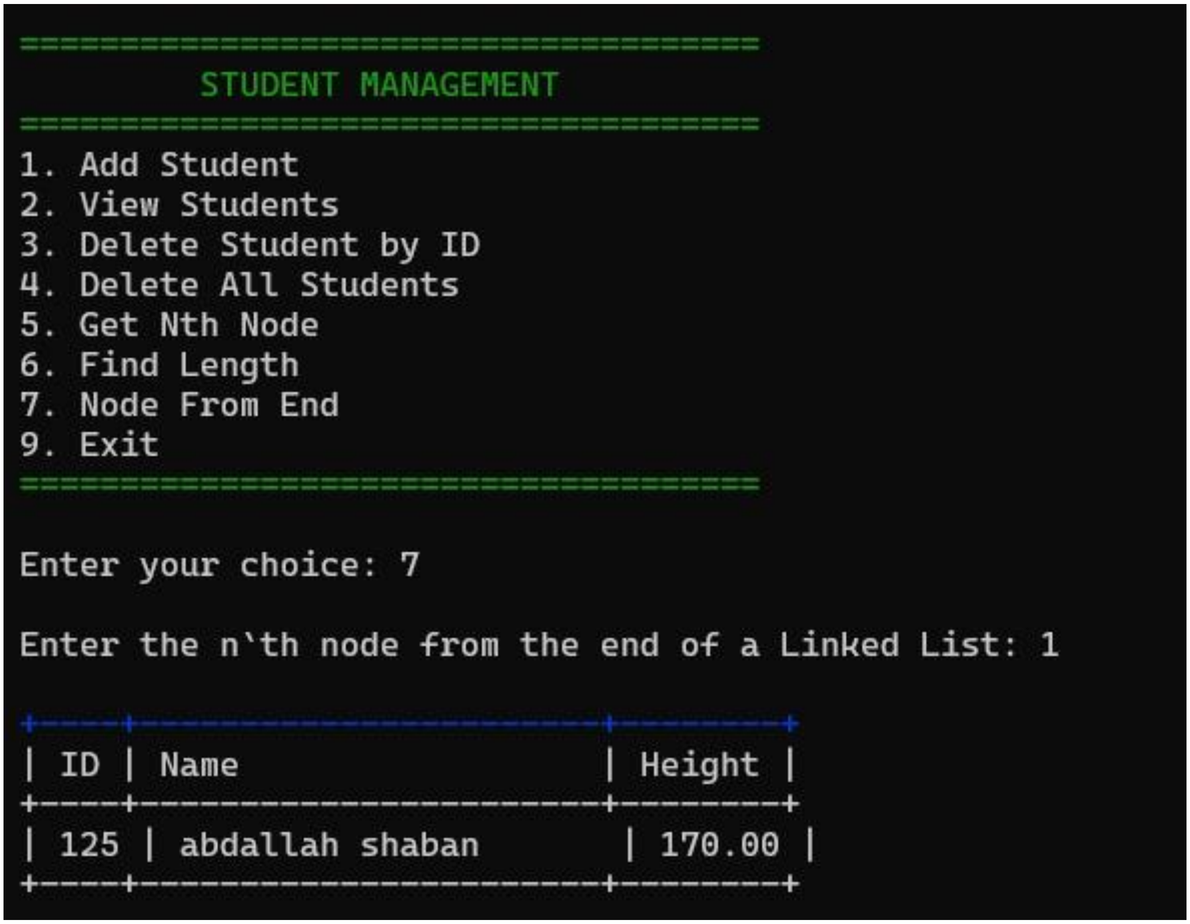
10. Function to Find the Nth Node from the End



```c
void NodeFromEnd(void) {
    // Implementation
}
```

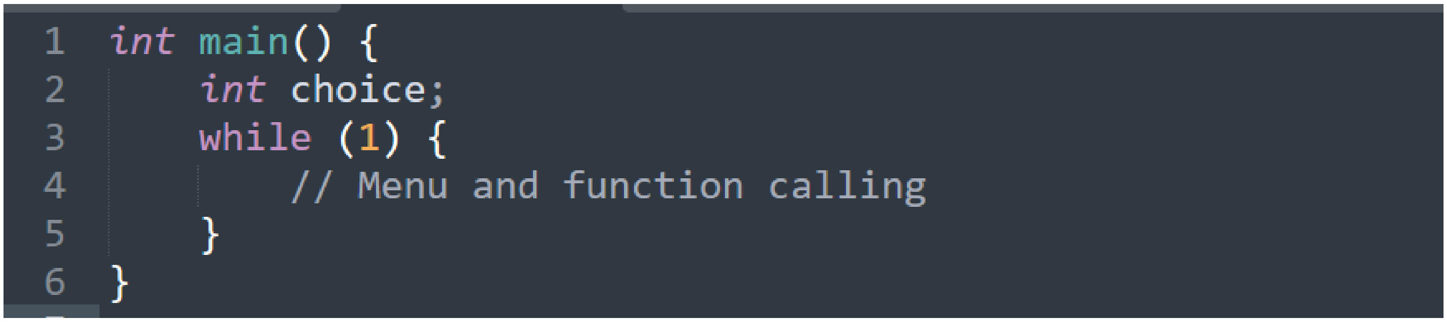
Explanation:

- This function finds the node that is n positions from the end of the list. It uses two pointers to achieve this, one that leads and another that trails.



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 7

Enter the n`th node from the end of a Linked List: 1

+----+-----------------------+--------+
| ID | Name                  | Height |
+----+-----------------------+--------+
| 125 | abdallah shaban        | 170.00 |
+----+-----------------------+--------+
```

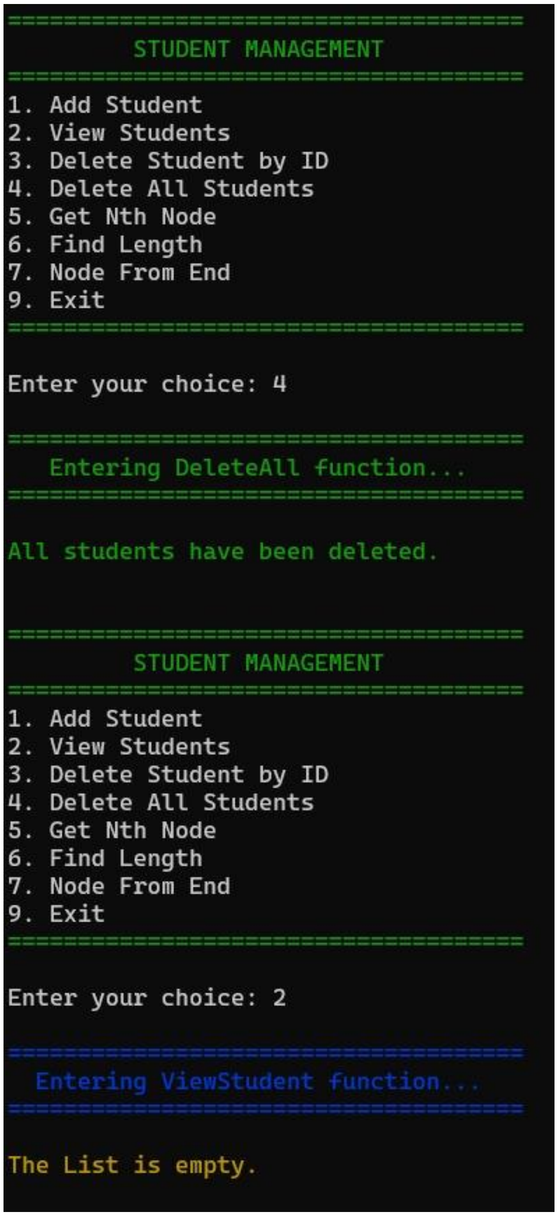
11. Main Function



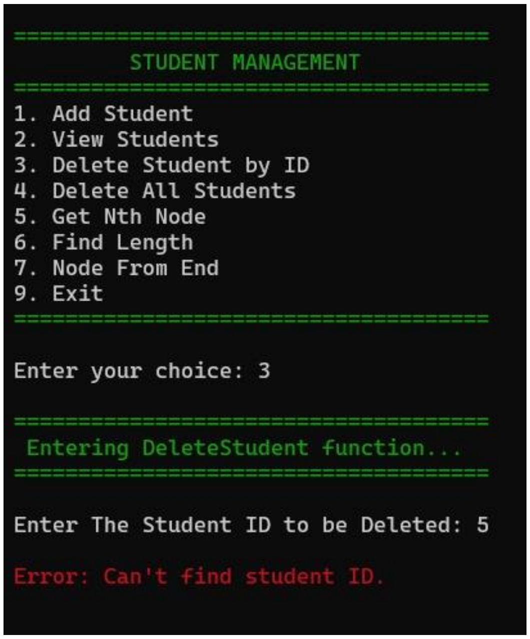
```c
int main() {
    int choice;
    while (1) {
        // Menu and function calling
    }
}
```

Explanation:

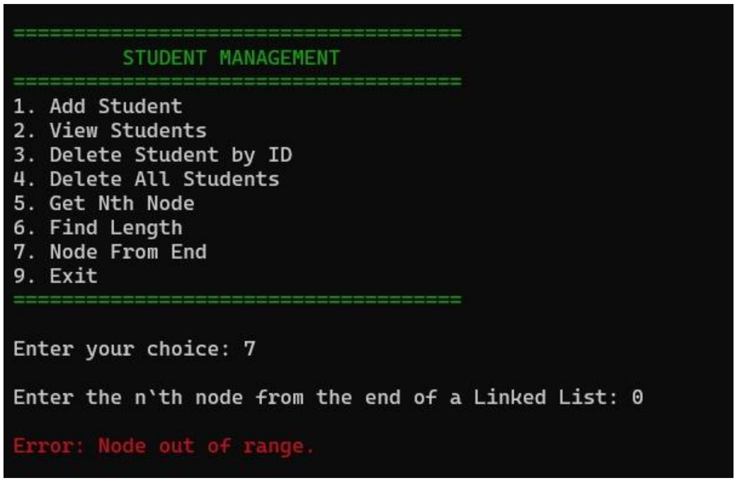
- The main function presents a menu to the user and calls the corresponding function based on the user's choice. It uses an array of function pointers to streamline function calls and manages memory cleanup before exiting.



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 4

=====================================
   Entering DeleteAll function...
=====================================

All students have been deleted.


=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 2

=====================================
  Entering ViewStudent function...
=====================================

The List is empty.
```



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 3

=====================================
 Entering DeleteStudent function...
=====================================

Enter The Student ID to be Deleted: 5

Error: Can't find student ID.
```



```
=====================================
         STUDENT MANAGEMENT
=====================================
1. Add Student
2. View Students
3. Delete Student by ID
4. Delete All Students
5. Get Nth Node
6. Find Length
7. Node From End
9. Exit
=====================================

Enter your choice: 7

Enter the n`th node from the end of a Linked List: 0

Error: Node out of range.
```