Project Title: "Employee Management System using Linked Lists"

This C program implements a basic Employee Management System using linked lists. The program allows users to perform various operations such as adding, deleting, searching, and displaying employee information. When the program starts, the globally defined head variable has a NULL value. This variable represents the head of our linked list.

The main function presents a menu to the user and calls the relevant function according to the user's selection. After the user selects an option from the menu, the program calls the function that performs the relevant action.

• If the user wants to add an employee, the addEmployee function is called. This function creates a new Employee structure and obtains the necessary information from the user. Using the checkDuplicate function, it is checked whether there is another employee with the same name. If there is an employee with the same name, the operation fails and the newly created Employee structure is released to prevent memory leaks. Otherwise, the new employee is added to the linked list and the head is updated.

• If the user wants to delete an employee, the deleteEmployee function is called. The name of the employee to be deleted is obtained from the user. The linked list is scanned and the employee to be deleted is found. If found, this worker is removed from the linked list and memory is freed.

• If the user wants to call an employee, the searchEmployee function is called. The name of the employee to be called is obtained from the user. The linked list is scanned and if an employee is found, their information is printed on the screen.

• The displayEmployees function is called to list all employees. The linked list is scanned and each employee's information is printed on the screen.

• The checkDuplicate function takes the name of the employee to be added and checks whether there is a previously added employee with this name. If there is an employee with the same name, the value 1 is returned; otherwise, 0 is returned.

• freeMemory function is used to release memory space of a worker.

• If necessary, the validateInput function can be used to validate inputs received from the user.

• When the user wants to terminate the program, the memory of all workers in the linked list is released and the program is closed.