Aim: Practicing with linked-list operations.

Write a program that uses a linked-list to illustrate a firm’s employee-list. The list should be implemented as a single linked-list structure. This employee-list is empty at first. Then, during the firm’s development, the list will point the nodes as the records of the employee’s information. Each node, as employee information, is a structure that holds the name, the phone number and the status information (manager, project leader, engineer, technician, secretary) of the related employee. Remember to use dynamic allocation concepts for the employee nodes and implement your program suitable to insert, search and print the employee nodes. An example design about the program is given below:

```c
struct Employee{
  // information data
  // a single self referential pointer to a similar structure
};

struct EmployeeList{
  // a pointer to the first Employee node
  // a pointer to the last Employee node
  // an auxiliary data to count the total number of Employee nodes
};

// example prototypes of the list functions
struct EmployeeList* insertEmployee(struct EmployeeList*, struct Employee*);
  // insertEmployee() inserts a new node to the list and returns the list
int searchEmployee(struct EmployeeList*, char* name);
  // searchEmployee() searches for the employee info in the list
  // returns 1 if the employee with the given “name” is found, or returns 0 otherwise
int numberOfEmployee(struct EmployeeList*);
  // returns the number of nodes in the list
void printEmployee(struct EmployeeList*);
  // iteratively prints all Employee info in the list
void recursivePrintEmployee(struct EmployeeList*);
  // recursively prints all Employee info
int isEmpty(struct EmployeeList*);
  // isEmpty() is an auxiliary function that returns 1 if the given list is empty, or 0 otherwise

int main()
  // make appropriate tests for your implementation, such as shown below:
  // create an employee list
  // insert some employee nodes into the list and get the number of nodes occasionally
  // search for a specific employee with its name info
  // print all the employee info available in the list
  return 0;
```