List of Employees

In this assignment, you have three given files (attached to the assignment):

- "employee.h" contains the structure required for this assignment and prototypes of all functions.
- "main.c" contains different test cases that you need to verify.
- "employee.c" will contain the code that you must fill.

The aim of the code is to implement a dynamic list of employees. The struct of the EMPLOYEE and PAYMENT is provided in the file "employee.h" as follows:

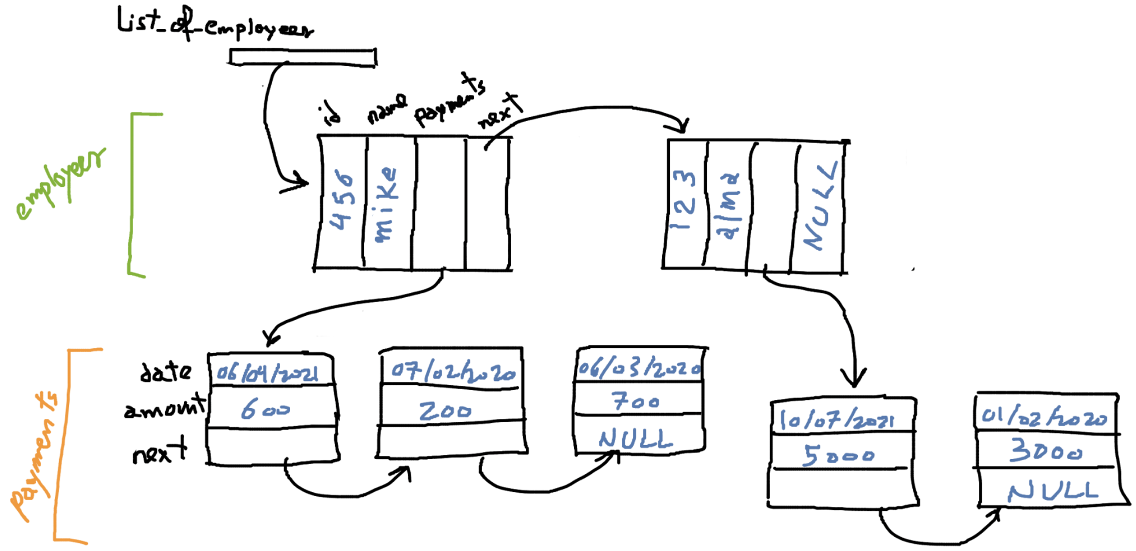
```
typedef struct payment{
    char* date;         //  MM/DD/YY as text
    float amount;
    struct payment* next;
}PAYMENT;

typedef struct employee{
    int id;
    char* name;
    PAYMENT* payments;
    struct employee* next;
}EMPLOYEE;
```

A global variable (list_of_employees) was created to point to the head of the list (this was defined in the file "employee.c").

```
EMPLOYEE* list_of_employees=NULL;
```

The list should look like this (for example):



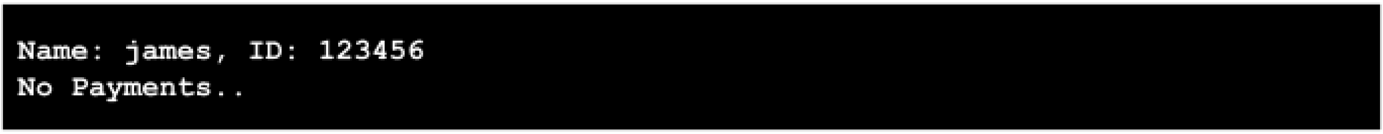
You need to implement the code of the following functions (in the file "employee.c"):

- **void printEmployee(EMPLOYEE e);**
    - This function takes an EMPLOYEE record as input and prints the details of that employee and his/her payments.
- **void printEmployees();**
    - This function should print all employees in the linked list with their payments.
- **void addEmployee(int e_id, char* e_name);**

    - This function takes a number (e_id) and text (e_name). It should create an employee and add that employee at the beginning of the linked list.
- **int addPayment(int e_id, char* p_date, float p_amount);**
    - This function takes an employee id (e_id), date and an (amount). It should create a payment with the given amount and date and add that to the list of payments of the employee with the id (e_id). This should be added at the beginning of the payments linked list (Assume that the employee id is unique i.e. there will be no two employees with the same id in the linked list).
    - The function should return 1 if the employee was in the list and the payment was added correctly or returns 0 if the employee was not found.
- **int deleteEmployee(int e_id, char* e_name);**
    - This function takes an employee id (e_id), and text (e_name). It should delete the employee who has that id (e_id) from the list of employees (Assume that the employee id is unique i.e. there will be no two employees with the same id in the linked list).
    - The function should return 1 if the employee was in the list deletion process was done correctly or returns 0 if the employee was not found.

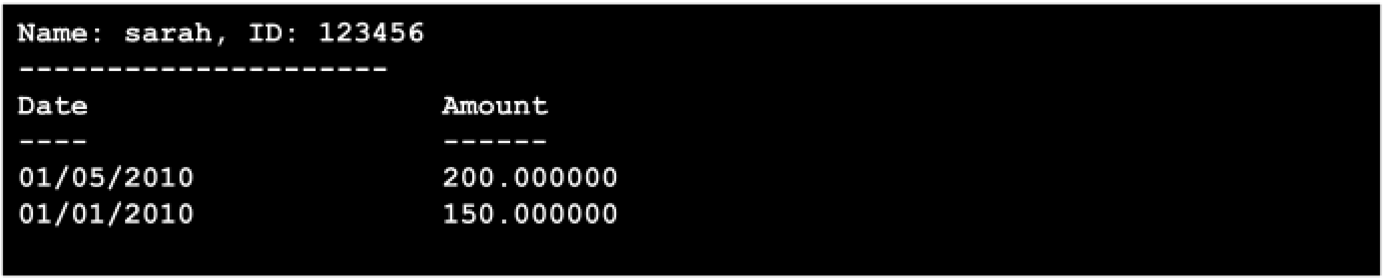
In "main.c", you have given different test cases that can help you test your code. Be sure that you get the expected output when you run each test case as follows:

TestCase -1 expected output:



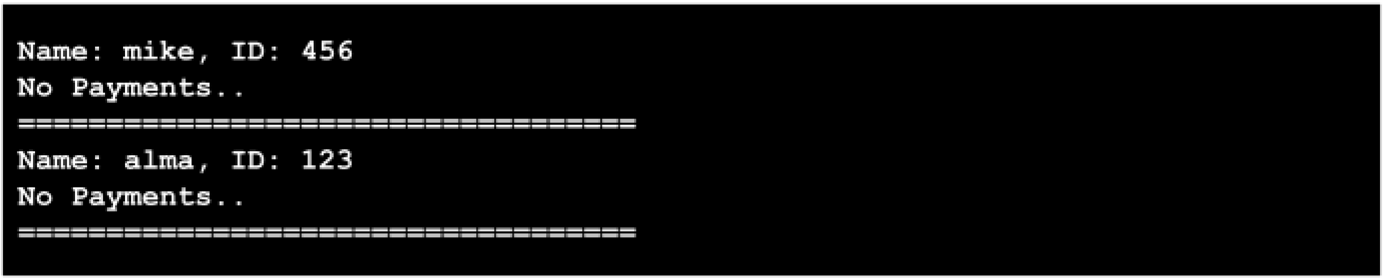
```
Name: james, ID: 123456
No Payments..
```

TestCase -2 expected output:



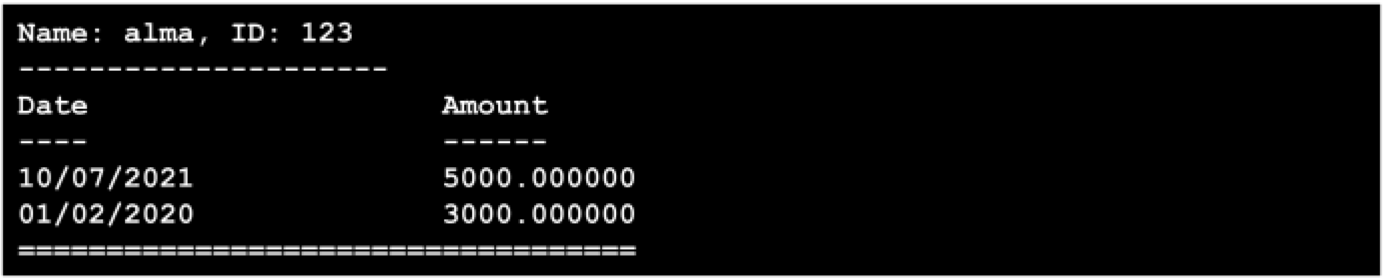
```
Name: sarah, ID: 123456
---------------------
Date                    Amount
----                    ------
01/05/2010              200.000000
01/01/2010              150.000000
```

TestCase -3 expected output:



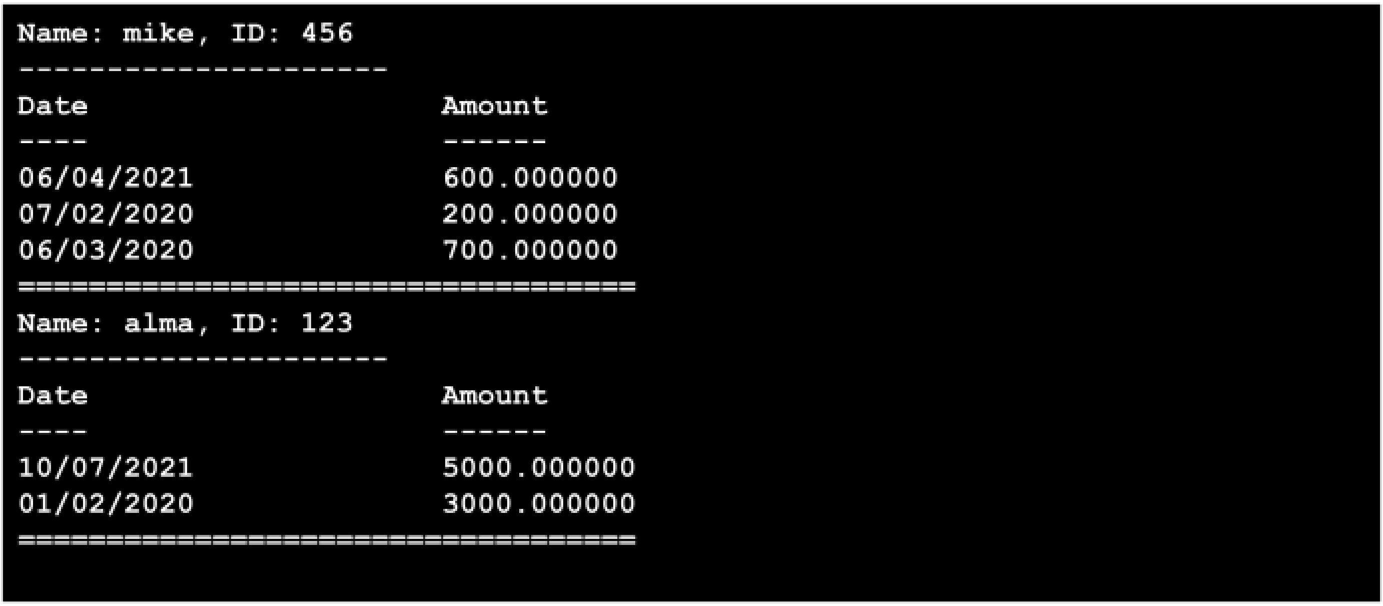
```
Name: mike, ID: 456
No Payments..
===================================
Name: alma, ID: 123
No Payments..
===================================
```

TestCase -4 expected output:



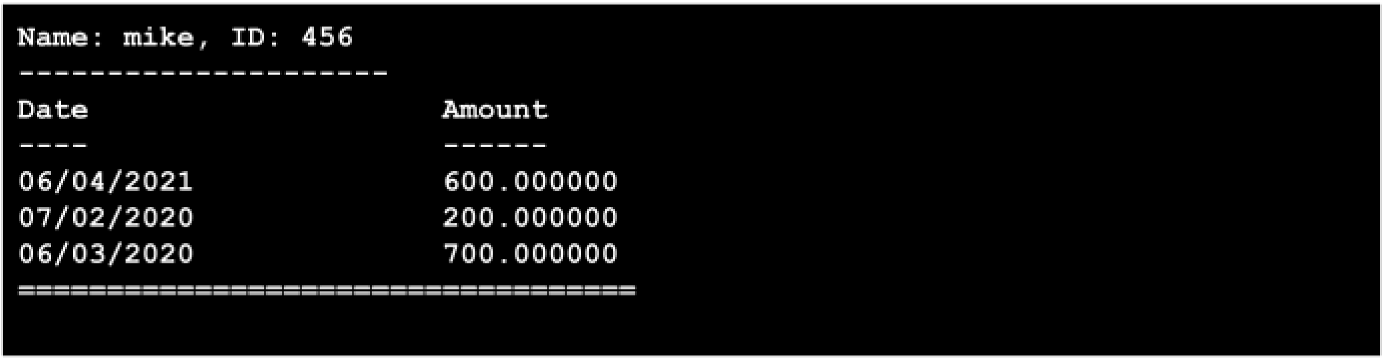
```
Name: alma, ID: 123
---------------------
Date                    Amount
----                    ------
10/07/2021              5000.000000
01/02/2020              3000.000000
===================================
```

TestCase -5 expected output:



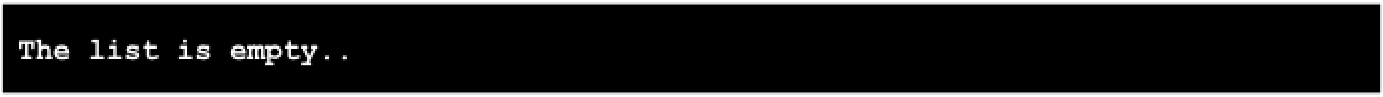
```
Name: mike, ID: 456
---------------------
Date                    Amount
----                    ------
06/04/2021              600.000000
07/02/2020              200.000000
06/03/2020              700.000000
===================================
Name: alma, ID: 123
---------------------
Date                    Amount
----                    ------
10/07/2021              5000.000000
01/02/2020              3000.000000
===================================
```

TestCase -6 expected output:



```
Name: mike, ID: 456
---------------------
Date                    Amount
----                    ------
06/04/2021              600.000000
07/02/2020              200.000000
06/03/2020              700.000000
===================================
```

TestCase -7 expected output:



```
The list is empty..
```

Important Notes:

- It is so important to submit a working program (Non-working applications will not be considered).
- You must submit one file only (“employee.c” that includes all your code).