Write the following functions and the main function for list processing for lists of nodes of the type node as follows.

Write a C program for the task as described below. Save the file as **A09_Roll_Number.c**. Build and run to check your program. Upload the .c file for the assignment in MS teams.

1. Define a node in a linked list as follows:

```
typedef struct _node {
        int key;
        char *name;
        struct _node *next;
} node;
```

2. Write a function node *createlist(int size) to create a list (with at least 2 nodes, $size \geq 2$), that returns a pointer to the first node of the list.

3. Also write a function void printlist (node *list) to print the list pointed to by the pointer list.

4. After creating n lists by calling the function createlist as many times, where n is given by the user, you must print all the n lists by using the printlist function as many times.

5. Write a function to concatenate the n lists to create one single list. This function must take as input an array of pointers that has the pointers to the first nodes of the n lists. It must return the pointer to the first node of the concatenated list.

6. Write functions also for inserting and deleting nodes; for insertion, insert at the end of the list. For deletion, delete the node(s) that have a certain user provided key value.

7. Destruction of a list may be done calling a function destroylist.

8. The main function must provide a menu.

Sample Output: Here is one sample output:

```
    Please enter n: 2

    Populate list 1:
    Enter name 1: Ram
key 10
    Enter name 2: Fatima
key 20
Populate list 2:
    Enter name 3: Ramu
key 30
    Enter name 4: Rohim
key 40


    Printing list 1:
```

Key: 10 Name: Ram
Key: 20 Name: Fatima

Printing list 2:

Key: 30 Name: Ramu
Key: 40 Name: Rohim

Printing concatenated list:

Key: 10 Name: Ram
Key: 20 Name: Fatima
Key: 30 Name: Ramu
Key: 40 Name: Rohim

Inserting new node $\{50, \mathsf{Martin}\}$

Printing new list:

Key: 10 Name: Ram
Key: 20 Name: Fatima
Key: 30 Name: Ramu
Key: 40 Name: Rohim
Key: 50 Name: Martin

Deleting node $\{30, \mathsf{Ramu}\}$

Printing new list:

Key: 10 Name: Ram
Key: 20 Name: Fatima
Key: 40 Name: Rohim
Key: 50 Name: Martin