NAME: JERRY DAVID R (192424401)

COURSE NAME: DATA STRUCTURES FOR MODERN COMPUTING SYSTEMS

COURSE CODE: CSA0302

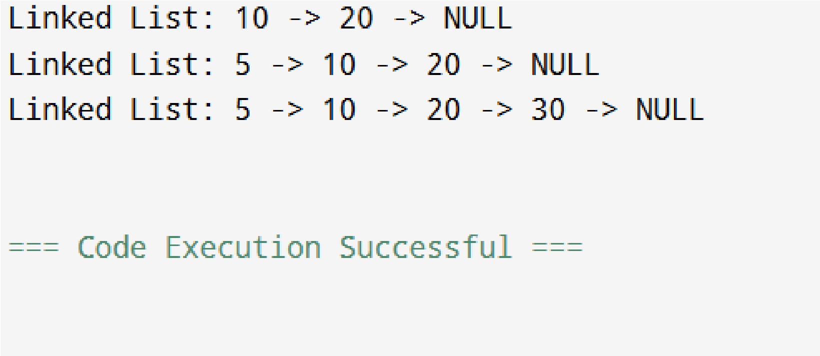
Experiment 11: Linked List

Code:

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
   int data;
   struct Node* next;
};
struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   if (newNode == NULL) {
      printf("Memory allocation failed!\n");
      exit(1);
   }
   newNode->data = data;
   newNode->next = NULL;
   return newNode;
}
void insertAtBeginning(struct Node** headRef, int data) {
   struct Node* newNode = createNode(data);
   newNode->next = *headRef;
   *headRef = newNode;
}
void insertAtEnd(struct Node** headRef, int data) {
   struct Node* newNode = createNode(data);
   if (*headRef == NULL) {
      *headRef = newNode;
```

```
        return;
    }
    struct Node* last = *headRef;
    while (last->next != NULL) {
        last = last->next;
    }
    last->next = newNode;
}
void printList(struct Node* node) {
    printf("Linked List: ");
    while (node != NULL) {
        printf("%d -> ", node->data);
        node = node->next;
    }
    printf("NULL\n");
}
void freeList(struct Node* head) {
   struct Node* tmp;
   while (head != NULL) {
       tmp = head;
       head = head->next;
       free(tmp);
    }
}
int main() {
    struct Node* head = NULL;
    insertAtEnd(&head, 10);
    insertAtEnd(&head, 20);
    printList(head);
    insertAtBeginning(&head, 5);
    printList(head);
    insertAtEnd(&head, 30);

    printList(head);

    freeList(head);

    return 0;

}
```

Output:



```
Linked List: 10 -> 20 -> NULL
Linked List: 5 -> 10 -> 20 -> NULL
Linked List: 5 -> 10 -> 20 -> 30 -> NULL


=== Code Execution Successful ===
```