2.Implement queue using singly linked list.

Shashank Patel C J

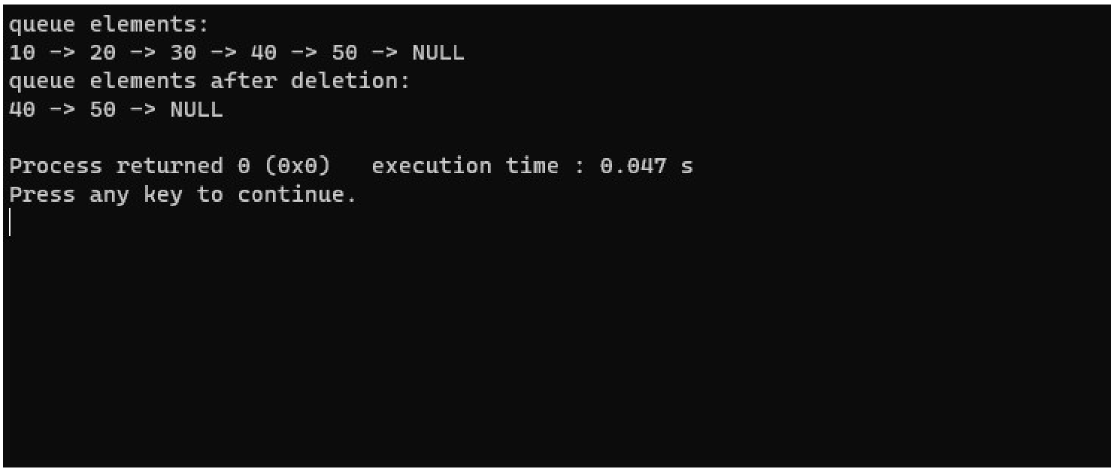
1BM22CS255

```c
#include <stdio.h>
#include <stdlib.h>

struct Node {
   int data;
   struct Node* next;
};
void insertAtEnd(struct Node** head, int value)
{
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   struct Node* temp = *head;
   newNode->data = value;
   newNode->next = NULL;

   if (*head == NULL) {
      *head = newNode;
      return;
   }

   while (temp->next != NULL) {
      temp = temp->next;
   }
```

```
    temp->next = newNode;
}
void deleteAtBeginning(struct Node** head) {
    if (*head == NULL) {
        printf("Linked list is already empty.\n");
        return;
    }
    struct Node* temp = *head;
    *head = (*head)->next;
    free(temp);
}
void display(struct Node* head)
{
    struct Node* temp = head;

    if (temp == NULL) {
        printf("Linked list is empty.\n");
        return;
    }

    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }

    printf("NULL\n");
```

```
}
int main()
{
   struct Node* head=NULL;
   insertAtEnd(&head,10);
   insertAtEnd(&head,20);
   insertAtEnd(&head,30);
   insertAtEnd(&head,40);
   insertAtEnd(&head,50);
   printf("queue elements:\n");
   display(head);
   deleteAtBeginning(&head);
   deleteAtBeginning(&head);
   deleteAtBeginning(&head);
   printf("queue elements after deletion:\n");
   display(head);
   return 0;
}
```

Output:



```
queue elements:
10 -> 20 -> 30 -> 40 -> 50 -> NULL
queue elements after deletion:
40 -> 50 -> NULL

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```