DS PRACTICAL 08 [B]

AIM: Implement a Queue using Linked List and perform the Queue operations: Enqueue, Dequeue and Print using Menu Driver Program such as 1.Add, 2.Delete and 3.Print and 4. Exit.

PROGRAM :

```
#include<stdio.h>

#include<stdlib.h>


//Structure of the node

struct node{

   int data;

   struct node* next;

};

int data;

struct node* front = NULL;

struct node* rear = NULL;


//Inserting data in queue.(Enqueue function):

int enqueue(){

   //Creating the node first

   struct node* p;

   p = (struct node*)malloc(sizeof(struct node));

   if(p == NULL){

      //Checking the queue is overflow or not

      printf("The Queue is overflow\n");

   }

   printf("Enter the data:\t");

   scanf("%d", &p->data);

   p->next = NULL; // Initialize new node's next to NULL


   if (front == NULL && rear == NULL)

   {
```

```
        // First element in queue
        front = rear = p;
    }
    else
    {
        // Add to the end of the queue
        rear->next = p;
        rear = p;
    }

    return 0;
}

// Deleting data in queue.(Dequeue function):
int dequeue(){
    struct node* p;
    if(front == NULL && rear == NULL){
        printf("The Queue is underflow\n");
    }
    else
    {
        struct node *p = front;
        printf("The deleting data is %d\n", front->data);
        front = front->next;

        if (front == NULL)
        {
            // If queue becomes empty, update rear to NULL
            rear = NULL;
        }
```

```
        free(p);

    }

    return 0;
}

void display(){
    struct node* display;
    display = front;
    if(front == NULL){
        printf("The Queue is empty can not print the element.\n\n");
    }else{
    printf("The data in the Queue:\t\n");
    while(display != NULL){
        printf("%d\t", display -> data);
        display = display -> next;
    }
    printf("\n" );
    }
}

int main(){
    int choice;
    printf("Queue Implementation using Linked List\n");
    printf("Choices\n1.Enqueue\t2.Dequeue\t3.Print\t4.Exit\n");
    do
    {   printf("Enter the choice:\t");
        scanf("%d",&choice);

        switch (choice)
```

```
        {
        case 1:

            enqueue();
            break;
        case 2:
            dequeue();
            break;
        case 3:
            display();
            break;
        case 4:
            printf("You exit the program successfully.\n");
            break;
        default:
        printf("Please enter valid choice as mention\n");
            break;
        }

    } while (choice != 4);

    return 0;
}
```

OUTPUT

```
PS C:\Users\chuna> g++ p8b.c
PS C:\Users\chuna> ./a.exe
Queue Implementation using Linked List
Choices
1.Enqueue       2.Dequeue       3.Print 4.Exit
Enter the choice:       1
Enter the data: 12
Enter the choice:       1
Enter the data: 23
Enter the choice:       1
Enter the data: 34
Enter the choice:       1
Enter the data: 45
Enter the choice:       1
Enter the data: 56
Enter the choice:       2
The deleting data is 12
Enter the choice:       3
The data in the Queue:
23      34      45      56
Enter the choice:       4
You exit the program successfully.
PS C:\Users\chuna>
```

GITHUB LINK : https://github.com/Nishikant-Chunarkar/DATA_STRUCTURE_PRACTICAL