QUEUE USING LINKED LIST

```
     /****    Program to Implement Queue using Linked List    ****/

#include<stdio.h>

struct node
{
     int info;
     struct node *link;
}*front = NULL, *rear = NULL;

void insert();
void delet();
void display();

int item;

main()
{
     int ch;

     do
     {
          printf("\n\n1.\tInsert\n2.\tDelete\n3.\tDisplay\n4.\tExit\n");
          printf("\nEnter your choice: ");
          scanf("%d", &ch);

          switch(ch)
          {
               case 1:
                    insert();
                    break;

               case 2:
                    delet();
                    break;

               case 3:
                    display();
                    break;

               case 4:
                    exit(0);
```

```
            default:
                printf("\n\nInvalid choice. Please try again...\n");
        }
    } while(1);
    getch();
}

void insert()
{
    printf("\n\nEnter ITEM: ");
    scanf("%d", &item);

    if(rear == NULL)
    {
        rear = (struct node *)malloc(sizeof(struct node));
        rear->info = item;
        rear->link = NULL;
        front = rear;
    }
    else
    {
        rear->link = (struct node *)malloc(sizeof(struct node));
        rear = rear->link;
        rear->info = item;
        rear->link = NULL;
    }
}

void delet()
{
    struct node *ptr;

    if(front == NULL)
        printf("\n\nQueue is empty.\n");
    else
    {
        ptr = front;
        item = front->info;
        front = front->link;
        free(ptr);
        printf("\nItem deleted: %d\n", item);

        if(front == NULL)
            rear = NULL;
    }
}
```

```
void display()
{
     struct node *ptr = front;

     if(rear == NULL)
          printf("\n\nQueue is empty.\n");
     else
     {
          printf("\n\n");

          while(ptr != NULL)
          {
               printf("%d\t",ptr->info);
               ptr = ptr->link;
          }
     }
}
```