**C PROGRAM 1:**

**Write a C program to implement simple queue using SLL.**

#include <stdio.h>

#include <stdlib.h>

#define TRUE 1

#define FALSE 0

#define FULL 10

struct node

{

int data;

struct node \*next;

};

typedef struct node node;

struct queue

{

int count;

node \*front;

node \*rear;

};

typedef struct queue queue;

void initialize(queue \*q)

{

q->count = 0;

q->front = NULL;

q->rear = NULL;

}

int isempty(queue \*q)

{

return (q->rear == NULL);

}

void enqueue(queue \*q, int value)

{

if (q->count < FULL)

{

node \*tmp;

tmp = malloc(sizeof(node));

tmp->data = value;

tmp->next = NULL;

if(!isempty(q))

{

q->rear->next = tmp;

q->rear = tmp;

}

else

{

q->front = q->rear = tmp;

}

q->count++;

}

else

{

printf("List is full\n");

}

}

int dequeue(queue \*q)

{

node \*tmp;

int n = q->front->data;

tmp = q->front;

q->front = q->front->next;

q->count--;

free(tmp);

return(n);

}

void display(node \*head)

{

if(head == NULL)

{

printf("NULL\n");

}

else

{

printf("%d\n", head -> data);

display(head->next);

}

}

int main()

{

queue \*q;

q = malloc(sizeof(queue));

initialize(q);

enqueue(q,10);

enqueue(q,20);

enqueue(q,30);

printf("Queue before dequeue\n");

display(q->front);

dequeue(q);

printf("Queue after dequeue\n");

display(q->front);

return 0;

}

**OUTPUT:**

