**EXPERMINET-9**

**Aim:** Create a Linear Queue using Linked List and implement different operations such as Insert, Delete, and Display the queue elements.

**ALGORITHM:**

**SOURCE CODE:**

#include <iostream>

using namespace std;

struct node {

int data;

struct node \*next;

};

struct node\* front = NULL;

struct node\* rear = NULL;

struct node\* temp;

void Insert() {

int val;

cout<<"Insert the element in queue : ";

cin>>val;

if (rear == NULL) {

rear = (struct node \*)malloc(sizeof(struct node));

rear->next = NULL;

rear->data = val;

front = rear;

} else {

temp=(struct node \*)malloc(sizeof(struct node));

rear->next = temp;

temp->data = val;

temp->next = NULL;

rear = temp;}}

void Delete() {

temp = front;

if (front == NULL) {

cout<<"Underflow"<<endl;

return;}

else

if (temp->next != NULL) {

temp = temp->next;

cout<<"Element deleted from queue is : "<<front->data<<endl;

free(front);

front = temp;

} else {

cout<<"Element deleted from queue is : "<<front->data<<endl;

free(front);

front = NULL;

rear = NULL;}}

void Display() {

temp = front;

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

cout<<"Queue is empty"<<endl;

return;}

cout<<"Queue elements are: ";

while (temp != NULL) {

cout<<temp->data<<" ";

temp = temp->next;}

cout<<endl;

}

int main() {

int ch;

cout<<"1 Insert"<<endl;

cout<<"2) Delete"<<endl;

cout<<"3) Display"<<endl;

cout<<"4) Exit"<<endl;

do {

cout<<"Enter your choice : ";

cin>>ch;

switch (ch) {

case 1: Insert();

break;

case 2: Delete();

break;

case 3: Display();

break;

case 4: cout<<"Exit"<<endl;

break;

default: cout<<"Invalid choice"<<endl;}

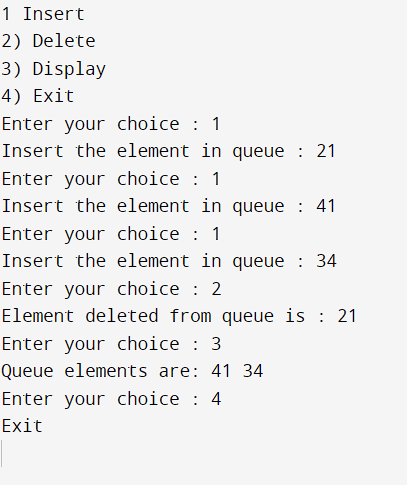
} while(ch!=4);

return 0;}

**EXPERMINET-9**

**Aim:** Create a Linear Queue using Linked List and implement different operations such as Insert, Delete, and Display the queue elements.

**OUTPUT:**

****