Practical-08

8B . 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.

//queue using linked list

#include <stdio.h>

#include <stdlib.h>

// Define a node structure

struct Node {

int data;

struct Node\* next;

};

// Function prototypes

void enqueue(struct Node\*\* front, struct Node\*\* rear, int value);

void dequeue(struct Node\*\* front, struct Node\*\* rear);

void display(struct Node\* front);

void menu();

int main() {

struct Node\* front = NULL;

struct Node\* rear = NULL;

int choice, value;

while (1) {

menu();

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter the value to enqueue: ");

scanf("%d", &value);

enqueue(&front, &rear, value);

break;

case 2:

dequeue(&front, &rear);

break;

case 3:

display(front);

break;

case 4:

printf("Exiting program.\n");

exit(0);

default:

printf("Invalid choice, please try again.\n");

}

}

return 0;

}

// Function to display the menu

void menu() {

printf("\nQueue using Linked List\n");

printf("1. Enqueue\n");

printf("2. Dequeue\n");

printf("3. Display\n");

printf("4. Exit\n");

}

// Function to enqueue an element

void enqueue(struct Node\*\* front, struct Node\*\* rear, int value) {

struct Node\* newNode = (struct Node\*)malloc(sizeof(struct Node));

if (!newNode) {

printf("Memory allocation error\n");

return;

}

newNode->data = value;

newNode->next = NULL;

if (\*rear == NULL) {

\*front = \*rear = newNode;

} else {

(\*rear)->next = newNode;

\*rear = newNode;

}

printf("Enqueued %d\n", value);

}

// Function to dequeue an element

void dequeue(struct Node\*\* front, struct Node\*\* rear) {

if (\*front == NULL) {

printf("Queue is empty\n");

return;

}

struct Node\* temp = \*front;

\*front = (\*front)->next;

if (\*front == NULL) {

\*rear = NULL;

}

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

free(temp);

}

// Function to display the queue

void display(struct Node\* front) {

if (front == NULL) {

printf("Queue is empty\n");

return;

}

struct Node\* temp = front;

printf("Queue elements: ");

while (temp != NULL) {

printf("%d ", temp->data);

temp = temp->next;

}

printf("\n");

}

Output: