Practical-08

8A . Implement a Queue 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.

// implement queue using array

#include <stdio.h>

#include <stdlib.h>

#define MAX 100

struct Queue {

int items[MAX];

int front, rear;

};

void initialize(struct Queue \*q) {

q->front = -1;

q->rear = -1;

}

int isFull(struct Queue \*q) {

return q->rear == MAX - 1;

}

int isEmpty(struct Queue \*q) {

return q->front == -1 || q->front > q->rear;

}

void enqueue(struct Queue \*q, int value) {

if (isFull(q)) {

printf("Queue is full!\n");

} else {

if (q->front == -1) q->front = 0;

q->rear++;

q->items[q->rear] = value;

printf("%d enqueued to queue\n", value);

}

}

void dequeue(struct Queue \*q) {

if (isEmpty(q)) {

printf("Queue is empty!\n");

} else {

printf("%d dequeued from queue\n", q->items[q->front]);

q->front++;

}

}

void display(struct Queue \*q) {

if (isEmpty(q)) {

printf("Queue is empty!\n");

} else {

printf("Queue elements are:\n");

for (int i = q->front; i <= q->rear; i++) {

printf("%d ", q->items[i]);

}

printf("\n");

}

}

int main() {

struct Queue q;

initialize(&q);

int choice, value;

while (1) {

printf("\nQueue Menu:\n");

printf("1. Enqueue\n");

printf("2. Dequeue\n");

printf("3. Display\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("Enter value to enqueue: ");

scanf("%d", &value);

enqueue(&q, value);

break;

case 2:

dequeue(&q);

break;

case 3:

display(&q);

break;

case 4:

exit(0);

default:

printf("Invalid choice! Please try again.\n");

}

}

return 0;

}

Output: