

## Practical 8[A]

Aim: 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.

Program:

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#define MAX 5 // Define the maximum size of the queue
```

```
// Define the Queue structure
```

```
struct Queue {
```

```
    int items[MAX];
```

```
    int front;
```

```
    int rear;
```

```
};
```

```
// Function to initialize the queue
```

```
void initializeQueue(struct Queue *q) {
```

```
    q->front = -1;
```

```
    q->rear = -1;
```

```
}
```

```
// Function to check if the queue is full
```

```
int isFull(struct Queue *q) {
```

```
    if (q->rear == MAX - 1) {
```

```
        return 1; // Queue is full
```

```
    }
```

```
    return 0;
```

```
}
```

```
// Function to check if the queue is empty
```

```
int isEmpty(struct Queue *q) {
```

```
    if (q->front == -1) {
```

```

        return 1; // Queue is empty
    }

    return 0;
}

// Function to add an item to the queue (Enqueue)
void enqueue(struct Queue *q, int value) {
    if (isFull(q)) {
        printf("Queue is full. Cannot enqueue %d\n", value);
        return;
    }
    if (q->front == -1) { // If the queue is empty, set front to 0
        q->front = 0;
    }
    q->rear++;
    q->items[q->rear] = value;
    printf("Enqueued %d to the queue\n", value);
}

// Function to remove an item from the queue (Dequeue)
void dequeue(struct Queue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty. Cannot dequeue\n");
        return;
    }
    int removedItem = q->items[q->front];
    printf("Dequeued %d from the queue\n", removedItem);
    q->front++;
    if (q->front > q->rear) { // If the queue is empty after dequeuing, reset to initial state
        q->front = q->rear = -1;
    }
}

// Function to print the queue contents

```

```

void printQueue(struct Queue *q) {
    if (isEmpty(q)) {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue contents: ");
    for (int i = q->front; i <= q->rear; i++) {
        printf("%d ", q->items[i]);
    }
    printf("\n");
}

```

// Function to display the menu and execute user operations

```

void menu() {
    struct Queue q;

    initializeQueue(&q); // Initialize the queue

    int choice, value;

    while (1) {
        // Display the menu
        printf("\nMenu:\n");
        printf("1. Add (Enqueue)\n");
        printf("2. Delete (Dequeue)\n");
        printf("3. Print Queue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                // Enqueue operation
                printf("Enter the value to enqueue: ");
                scanf("%d", &value);
                enqueue(&q, value);

```

```
break;
```

```
case 2:
```

```
    // Dequeue operation
```

```
    dequeue(&q);
```

```
    break;
```

```
case 3:
```

```
    // Print the queue
```

```
    printQueue(&q);
```

```
    break;
```

```
case 4:
```

```
    // Exit the program
```

```
    printf("Exiting the program.\n");
```

```
    exit(0);
```

```
default:
```

```
    printf("Invalid choice. Please enter a valid option.\n");
```

```
    }
```

```
    }
```

```
}
```

```
int main() {
```

```
    menu(); // Start the menu-driven program
```

```
    return 0;
```

```
}
```

```
File Edit Selection View Go Run Terminal Help c program
EXPLORER
> OPEN EDITORS
c PROGRAM
  a.exe
  bubble.c
  circular.c
  ex1.cpp
  exp.cpp
  exp.exe
  first.cpp
  gun.c
  hello.c
  hu.cpp
  linklist.c
  linklist1.c
  p.cpp
  pointer.c
  pointer2.c
  pr1a.c
  pr1b.c
  prac5b.c
  search.c
  second.cpp
  selection.c
  stac1.cpp
  stack.c
  stack.cpp
  string.c
  string.cpp
  today.c
  w.c
  w.exe
  x.c
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mathaw\OneDrive\Desktop\c program> .\a.exe

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 78
Enqueued 78 to the queue

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 45
Enqueued 45 to the queue

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
Enter the value to enqueue: 63
Enqueued 63 to the queue

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 3
Queue contents: 78 45 63

Menu:
1. Add (Enqueue)
2. Delete (Dequeue)
3. Print Queue
4. Exit
Enter your choice: 1
```

Github link :- <https://github.com/MayurThaware122/DSA>