Program - 06	
Title :	Write a Program to implement the operations of Queue using array.

Code:

```
#include <stdio.h>
#define MAX_SIZE 100

// Queue structure
struct Queue {
    int arr[MAX_SIZE];
    int front;
    int rear;
};

// Initialize a new queue
void initQueue(struct Queue* queue) {
    queue->front = -1;
    queue->rear = -1;
}

// Check if queue is empty
int isEmpty(struct Queue* queue) {
    return queue->front == -1;
}

// Check if queue is full
int isFull(struct Queue* queue) {
    return queue->rear == MAX_SIZE - 1;
}

// Add an integer to the back of the queue
void enqueue(struct Queue* queue, int data) {
    if (isFull(queue)) {
        printf("Queue is full!\n");
        return;
    }
    if (isEmpty(queue)) {
        queue->front = 0;
    }
    queue->rear++;
    queue->arr[queue->rear] = data;
}

// Remove an integer from the front of the queue
int dequeue(struct Queue* queue) {
    if (isEmpty(queue)) {
        printf("Queue is empty!\n");
        return 0;
    }
    int data = queue->arr[queue->front];
    if (queue->front == queue->rear) {
        queue->front = -1;
        queue->rear = -1;
    } else {
        queue->front++;
    }
    return data;
```

```
}

// Return the front integer of the queue without removing it
int peek(struct Queue* queue) {
    if (isEmpty(queue)) {
        printf("Queue is empty!\n");
        return 0;
    }
    return queue->arr[queue->front];
}

// Main function
int main() {
    struct Queue queue;
    initQueue(&queue);

    int choice, data;

    do {

        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Peek\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter data to enqueue: ");
                scanf("%d", &data);
                enqueue(&queue, data);
                break;
            case 2:
                printf("Dequeued data: %d\n", dequeue(&queue));
                break;
            case 3:
                printf("Front data: %d\n", peek(&queue));
                break;
            case 4:
                printf("Exiting program...\n");
                break;
            default:
                printf("Invalid choice!\n");
                break;
        }

    } while (choice != 4);

    return 0;
}
```

Output:

Enqueue

```
C:\Users\hp\Desktop\c program>aa
1. Enqueue
2. Dequeue
3. Peek
4. Exit
Enter your choice: 1
Enter data to enqueue: 23
1. Enqueue
2. Dequeue
3. Peek
4. Exit
Enter your choice: 1
Enter data to enqueue: 45
```

Dequeue

```
1. Enqueue
2. Dequeue
3. Peek
4. Exit
Enter your choice: 2
Dequeued data: 23
1. Enqueue
2. Dequeue
3. Peek
4. Exit
Enter your choice: 3
Front data: 45
```

Date : ___/____/_____

Teacher Sign ……