QUEUE IMPLEMENTATION USING ARRAY

```
#include <stdio.h>
#define MAX 5      // Maximum max of the queue
int queue[MAX];
int front = -1, rear = -1;
int isFull()       // Function to check if queue is full
{
   return (rear == MAX - 1);
}
int isEmpty()   // Function to check if queue is empty
{
   return (front == -1 || front > rear);
}
void enqueue(int data)
{
   if (isFull()) {
         printf("Queue Overflow! Cannot insert %d\n", data);
   }
 Else
 {
         if (front == -1)  // First element
         {front = 0;}
         queue[++rear] = data;
         printf("Inserted %d\n", data);
   }
}
void dequeue()
 {
   if (isEmpty()) {
         printf("Queue Underflow! Nothing to delete.\n");
   }
else {
         printf("Deleted %d\n", queue[front]);
         front++;
         }
}
 void peek()     // Peek: View front element without removing
{
   if (isEmpty()) {
         printf("Queue is empty. Nothing to peek.\n");
   } else {
         printf("Front element is: %d\n", queue[front]);
   }
}
```

```
void display()              // Display: Show all elements in the queue
 {
   if (isEmpty()) {
        printf("Queue is empty.\n");
   }
 else {
        printf("Queue elements: ");
        for (int i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
        }
        printf("\n");
   }
}
int main() {
   int choice, data;
   while (1) {
        printf("\n--- Queue Menu ---\n");
        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Display\n");
        printf("4. Peek\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
        case 1:
                printf("Enter data to insert: ");
                scanf("%d", &data);
                enqueue(data);
                break;
        case 2:
                dequeue();
                break;
        case 3:
                display();
                break;
        case 4:
                peek();
                break;
        case 5:
                printf("Exiting program.\n");
                return 0;
        default:
                printf("Invalid choice! Please try again.\n");
        }
   }
   return 0;}
}
```

QUEUE IMPLEMENTATION USING LINKED LIST

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

// Node structure
struct Node {
   int data;
   struct Node* next;
}* front = NULL, * rear = NULL;

// Enqueue operation
void enqueue(int data) {
   struct Node* temp = (struct Node*)malloc(sizeof(struct Node));
   if (temp == NULL) {
         printf("Memory allocation failed!\n");
         return;
   }
   temp->data = data;
   temp->next = NULL;

   if (rear == NULL) {  // Queue is empty
         front = rear = temp;
   } else {
         rear->next = temp;
         rear = temp;
   }
   printf("%d enqueued successfully.\n", data);
}

// Dequeue operation (no return data)
void dequeue() {
   if (front == NULL) {
         printf("Queue is empty! Cannot dequeue.\n");
         return;
   }

   struct Node* temp = front;
   printf("Dequeued element: %d\n", temp->data);
   front = front->next;

   if (front == NULL)  // Queue became empty
         rear = NULL;

   free(temp);
}
```

```
// Peek operation (no return)
void peek() {
    if (front == NULL) {
        printf("Queue is empty!\n");
    } else {
        printf("Front element: %d\n", front->data);
    }
}

// Display queue elements
void display() {
    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");
}

// Main function with interactive menu
int main() {
    int choice, data;
    while (1) {
        printf("\n---- Queue Menu ----\n");
        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Peek (Front Element)\n");
        printf("4. Display Queue\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
        case 1:
            printf("Enter data to enqueue: ");
            scanf("%d", &data);
            enqueue(data);
            break;
        case 2:
            dequeue();
            break;
        case 3:
                peek();
                break;
        case 4:
                display();
                break;
        case 5:
                printf("Exiting program.\n");
                exit(0);
        default:
                printf("Invalid choice! Please try again.\n");
        }
    }

    return 0;
}
```