

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

```
#define SIZE 5
```

```
int queue[SIZE];
```

```
int front = -1, rear = -1;
```

```
int isFull()
```

```
{
```

```
    return (rear + 1) % SIZE == front;
```

```
}
```

```
int isEmpty()
```

```
{
```

```
    return front == -1;
```

```
}
```

```
void enqueue(int data)
{
    if (isFull()) {
        printf("Queue overflow\n");
        return;
    }
    if (front == -1) {
        front = 0;
    }
    rear = (rear + 1) % SIZE;
    queue[rear] = data;
    printf("Element %d inserted\n", data);
}
```

```
int dequeue()
{
    if (isEmpty()) {
        printf("Queue underflow\n");
        return -1;
    }
    int data = queue[front];
    if (front == rear) {
        front = rear = -1;
    } else {
        front = (front + 1) % SIZE;
    }
    return data;
}
```

```

void traversal()
{
    if (isEmpty()) {
        printf("Queue is empty\n");
        return;
    }
    printf("Queue elements: ");
    int i = front;
    while (i != rear) {
        printf("%d ", queue[i]);
        i = (i + 1) % SIZE;
    }
    printf("%d\n", queue[rear]);
}

int main()
{
    int choice, data;

    while (1) {
        printf("\nCircular Queue Operations:\n");
        printf("1. Enqueue\n");
        printf("2. Dequeue\n");
        printf("3. Traversal Queue\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
    }
}

```

```
switch (choice) {  
    case 1:  
        printf("Enter value to enqueue: ");  
        scanf("%d", &data);  
        enqueue(data);  
        break;  
    case 2:  
        data = dequeue();  
        if (data != -1)  
            printf("Dequeued element: %d\n", data);  
        break;  
    case 3:  
        traversal();  
        break;  
    case 4:  
        return 0;  
    default:  
        printf("Invalid choice.\n");  
}  
}  
  
return 0;  
}
```