

```

#include <stdio.h>
#include <conio.h>
#define max 5
int front = -1;
int rear = -1;
int insert_rear();
int insert_front();
void display();
int deleteq_rear();
int deleteq_front();
int q[max];

int main() {
    int choice;
    clrscr();
    do {
        printf("\n **** Main Menu **** \n");
        printf("1. Insert From Rear\n");
        printf("2. Insert From Front\n");
        printf("3. Delete From Front\n");
        printf("4. Delete From Rear\n");
        printf("5. Display\n");
        printf("Enter your choice: ");

        scanf("%d", &choice);
        printf("\n");

        switch(choice) {
            case 1: insert_rear(); break;
            case 2: insert_front(); break;
            case 3: deleteq_front(); break;
            case 4: deleteq_rear(); break;
            case 5: display(); break;
            case 6: break;
        }
    } while (choice != 6);
    return 0;
}

int insert_rear() {
    int val;
    printf("Enter value: ");
    scanf("%d", &val);

```

```

if ((rear + 1) % max == front) {
    printf("Queue is full.");
    return 0;
} else if (rear == -1) {
    rear = front = 0;
    q[rear] = val;
    printf("Inserted successfully.");
    return val;
} else {
    rear = (rear + 1) % max;
    q[rear] = val;
    printf("Inserted successfully.");
    return val;
}
}

```

```

int insert_front() {
    int val;
    printf("Enter value: ");
    scanf("%d", &val);

    if ((rear + 1) % max == front) {
        printf("Queue is full.");
        return 0;
    } else if (front == -1) {
        rear = front = 0;
        q[front] = val;
        printf("Inserted successfully.");
        return val;
    } else {
        front = (front - 1 + max) % max;
        q[front] = val;
        printf("Inserted successfully.");
        return val;
    }
}

```

```

int deleteq_front() {
    int val;

    if (front == -1) {
        printf("Queue is empty.");
    } else if (front == rear) {
        val = q[front];
    }
}

```

```

        front = rear = -1;
        printf("Deleted value: %d", val);
        return val;
    } else {
        val = q[front];
        front = (front + 1) % max;
        printf("Deleted value: %d", val);
        return val;
    }
}

```

```

int deleteq_rear() {
    int val;

    if (rear == -1) {
        printf("Queue is empty.");
        return -1;
    } else if (front == rear) {
        val = q[rear];
        front = rear = -1;
        printf("Deleted value: %d", val);
        return val;
    } else {
        val = q[rear];
        rear = (rear - 1 + max) % max;
        printf("Deleted value: %d", val);
        return val;
    }
}

```

```

void display() {
    int i;
    if (front == -1) {
        printf("Queue is empty.");
    } else {
        printf("Queue is: ");
        for (i = front; i != rear; i = (i + 1) % max) {
            printf(" %d ", q[i]);
        }
        printf(" %d ", q[i]);
    }
}

```