Queue Implementation

```
#define N 5
int queue[N];
int front=-1,
int rear=-1;

void enque (x){
    if (rear==N-1) {
        printf("Queue overflow"); }
    else if (front==-1 && rear==-1){
        front=rear=0;
        queue[rear]=x;
    }
    else{
        rear++;
        queue[rear]=x;
    }

void deque (){
    if (front==-1 && rear==-1){
        printf("Queue is Empty");
    }
    else if (front==rear){
        front=rear=-1;
    }
    else {
        printf("%d", queue[front]);
        front++;
    }
```

```
void display(){
  if (front == -1 && rear == -1){
      printf("Queue is empty");
  }
  for(int i = front ; i <= rear ; i++){
      printf("%d", queue[i]);
  }
}

void peek(){
  if (front == -1 && rear == -1){
      printf("Queue is empty");
  }
  else {
      printf("front is %d", queue[front]);
  }
}

void main(){
  int c;
  printf("1. Enque
          2. Deque
          3. Display
          4. Peek
          5. Exit");

  scanf("Enter your choice:");
  scanf("%d", &c);
```

```
while (c != 5) {
    switch (c) {
        case 1:
        {int x;
        printf("Enter data:");
        scanf("%d", &x);
        enque(x);
        break;}
        case 2:
        deque();
        break;
        case 3;
        display();
        break;
        case 4:
        peek();
        break;
        default:
        printf("Invalid choice.");
        break;
    }
    printf("Enter next choice:");
    scanf("%d", &c);
}
printf("Exit!");
}
```

NG

O/P

1. Enque
2. Deque
3. Display
4. Peek
5. Exit

Enter a choice: 1
Enter data: 10
Enter choice: 1
Enter data: 20
Enter choice: 1
Enter data: 30
Enter choice: 1
Enter data: 40
Enter choice: 1
Enter data: 50
Enter choice: 1
Enter data: 60
Queue Overflow
Enter choice: 2
Deleted is 10
Enter choice: 3
20 30 40 50

Enter choice: 4
Front is 20

Enter choice: 5
Exit!

MG
13/10/25