## 6.1.1

```c
#include<stdio.h>
#define N 10
int ar[N];
int front=-1;
int rear=-1;

void enqueue(int x){
    if(rear==N-1){
        printf("Queue is full.\n");
    }
    if(front==-1&&rear==-1){
        front=0;
        rear=0;
    }
    else{
        rear++;
    }
    ar[rear]=x;
    printf("Successfully inserted.\n");
}
void dequeue(){
    int pop;
    if((rear==-1&&front==-1)||front>rear){
        printf("Queue is underflow.\n");
    }
    else{
        pop=ar[front];
        printf("Deleted element = %d\n",pop);
        front++;
```

```
}
void display(){
    int i;
    if((front==-1&&rear==-1)||(front>rear)){
        printf("Queue is empty.\n");
    }
    else{
        printf("Elements in the queue : ");
        for(i=front;i<=rear;i++){
            printf("%d ",ar[i]);
        }
        printf("\n");
    }
}
void isEmpty(){
    if((front==-1&&rear==-1)||(front>rear)){
        printf("Queue is empty.\n");
    }
    else{
        printf("Queue is not empty.\n");
    }
}
void size(){
    if(front==-1&&rear==-1){
        printf("Queue size : 0\n");
    }
    else{
        printf("Queue size : %d\n",rear-front+1);
    }
}
```

**6.1.2**

```c
#include<stdio.h>
#define N 5
int a[N];
int front =-1;
int rear=-1;
int i;
void enqueue(int x){
    if(front==-1&&rear==-1){
        front=0;
        rear=0;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
    else if((rear+1)%N==front){
        printf("Circular queue is overflow.\n");
    }
    else{
        rear=(rear+1)%N;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
}
void dequeue(){
    if(front==-1&&rear==-1){
        printf("Circular queue is underflow.\n");
    }
    else if(front==rear){
        printf("Deleted element = %d\n",a[front]);
        front=-1;
```

```
        rear=-1;
    }
    else{
        printf("Deleted element = %d\n",a[front]);
        front=(front+1)%N;
    }
}
void display(){
    if(front==-1){
        printf("Circular queue is empty.\n");
    }
    else{
        printf("Elements in the circular queue : ");
        i=front;

    do{
        printf("%d ",a[i]);
        i=(i+1)%N;
    }while(i!=rear);
    printf("%d \n",a[rear]);
    }
}
void isEmpty(){
    if(front==-1&&rear==-1){
        printf("Circular queue is empty.\n");
    }
    else{
        printf("Circular queue is not empty.\n");
    }
}
```

```
void size(){
    if(front==-1){
        printf("Circular queue size : 0\n");
    }
    else if(front<rear){
        printf("Circular queue size : %d\n",rear-front+1);
    }
    else if(front>rear){
        printf("Circular queue size : %d\n",rear+front);
    }
    else if(front==rear){
        printf("Circular queue size : %d\n",1);
    }
}
```

**6.1.3**

```c
#include<stdio.h>
#define N 10
int a[N];
int rear=-1,front=-1;
void inject(int x){
    if(rear==N-1){
        printf("Double ended queue is overflow\n");
    }
    else{
        rear++;
        a[rear]=x;
        if(front==-1){
            front=0;
        }
        printf("Successfully inserted at rear side.\n");
    }
}
void eject(){
    if(rear==-1){
        printf("Double ended queue is underflow.\n");
    }
    else{
        printf("Deleted element from the rear side = %d\n",a[rear]);
        if(front==rear){
            rear=-1;
            front=-1;
        }
        else{
            rear--;
        }
    }
}
void display(){
    int i;
    if(front==-1&&rear==-1){
        printf("Double ended queue is empty.\n");
    }
    else{
        printf("Elements in the double ended queue : ");
        for(i=front;i<=rear;i++){
            printf("%d ",a[i]);
        }
        printf("\n");
    }
}
```

## 6.2.1

```c
#include<stdio.h>
#define N 10
int a[N];
int front =-1;
int rear = -1;
void enqueue(int x){
    if(rear==N-1){
        printf("Queue is overflow.\n");
    }
    if(front==-1&&rear==-1){
        front=0;
        rear=0;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
    else{
        rear++;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
}
void dequeue(){
    int pop;
    if((front==-1&&rear==-1)||front>rear){
        printf("Queue is underflow.\n");
    }
    else{
        pop=a[front];
        printf("Deleted element = %d\n",pop);
```

```
        front++;
    }
}
void display(){
    int i;
    if(front==-1&&rear==-1){
        printf("Queue is empty.\n");
    }
    else{
        printf("Elements in the queue : ");
        for(i=front;i<=rear;i++){
            printf("%d ",a[i]);
        }
        printf("\n");
    }
}
```

## 6.2.2

```
#include<stdio.h>
#define N 10
int a[N];
int front=-1,rear=-1;
void enqueue(int x){
    if(front==-1&&rear==-1){
        front=0;
        rear=0;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
    else if((rear+1)%N==front){
        printf("Circular queue is overflow.\n");
    }
    else{
        rear=(rear+1)%N;
        a[rear]=x;
        printf("Successfully inserted.\n");
    }
}
void dequeue(){
    if(front==-1&&rear==-1){
        printf("Circular queue is underflow.\n");
    }
    else if(front==rear){
        printf("Deleted element = %d\n",a[front]);
        front=-1;
        rear=-1;
```

```
    }
    else{
        printf("Deleted element = %d\n",a[front]);
        front=(front+1)%N;
    }
  }
  void display(){
    int i;
    if(front==-1){
        printf("Circular queue is empty.\n");
    }
    else{
        printf("Elements in the circular queue : ");
        i=front;
        while(i!=rear){
            printf("%d ",a[i]);
            i=(i+1)%N;
        }
        printf("%d \n",a[rear]);
    }
  }
```

**6.2.3**

```c
#include<stdio.h>
int front=10,rear=9,arr[20];
void insertFront(int data){

    front--;
    arr[front]=data;
    printf("Inserted %d at front.\n",arr[front]);
}
void insertRear(int data){
    rear++;
    arr[rear]=data;
    printf("Inserted %d at rear.\n",arr[rear]);
}
void deleteFront(){
    if(front>rear){
        printf("Deque is empty.\n");
    }
    else{
        printf("Deleted %d from front.\n",arr[front]);
        front++;
    }
}
void deleteRear(){
    if(front>rear){
        printf("Deque is empty.\n");
    }
    else{
        printf("Deleted %d from rear.\n",arr[rear]);
        rear--;
    }
}
void display(){
    if(front>rear){
        printf("Deque is empty.\n");
    }
    else{
        printf("Deque: ");
        for(int i=front;i<=rear;i++){
            printf("%d ",arr[i]);
        }
        printf("\n");
    }
}
```