EXP 04

Program:

```
#include <stdio.h>

int MAX=5;
int deq[5];
int L=-1,R=-1;

void in_L();
void in_R();
void de_L();
void de_R();
void display();
void input_deque();
void output_deque();

void in_L(){
	if ((L==0 && R==MAX-1)|| L==R+1)
	{
		printf("\n Overflow");
		return;
	}
	int new;
	printf("\n Enter value to insert:");
	scanf("%d",&new);
	if (L==-1)    //if queue is initially empty
	{
		L=0;
		R=0;
	}
```

```
        else
        {
                if (L==0)
                        L=MAX-1;
                else
                        L--;
        deq[L]=new;
        }
}

void in_R(){
        if ((L==0 && R==MAX-1)|| L==R+1)   //check Overflow
        {
                printf("\n Overflow");
                return;
        }
        int new;
        printf("\n Enter value to insert:");
        scanf("%d",&new);
        if (L==-1)              //if queue is initially empty
        {
                L=0;
                R=0;
        }
        else{
                if (R==MAX-1)
                        R=0;
                else
                        R++;
        deq[R]=new;
        }
```

```
}

void de_L(){
	if (L==-1)
	{
		printf("\n Underflow");
		return;
	}
	if (L==R)
	{
		L=-1;
		R=-1;
	}
	else if(L==MAX-1)
		L==0;
	else
		L++;
}

void de_R(){
	if (L==-1)
	{
		printf("\n Underflow");
		return;
	}
	if (L==R)
	{
		L=-1;
		R=-1;
	}
	else if(R==0)
```

```
                R=MAX-1;
        else
                R--;
}

void display(){
        int Front=L, Rear=R;
        if (Front==-1){
                printf("\n Queue is Empty");
                return;
        }
        printf("\n Elements of Queue are: ");
        if (Front<=Rear)
        {
                while (Front<=Rear)
                {
                        printf("%d ", deq[Front]);
                        Front++;
                }
        }
        else{
                while(Front<=MAX-1)
                {
                        printf("%d ", deq[Front]);
                        Front++;
                }
        Front=0;
        while (Front<=Rear)
                {
                        printf("%d ", deq[Front]);
                        Front++;
```

```
                }
        }
}

void input_deque(){
        int ch;
        printf("\n Input Restricted Queue");
        do{
                printf("\n 1. Insert at Right\t 2. Delete at Right\t 3. Delete at Left\t 4.
Display\t 5. Exit");
                printf("\n Your choice:");
                scanf("%d", &ch);
                switch(ch){
                case 1: in_R();
                        break;
                case 2: de_R();
                        break;
                case 3: de_L();
                        break;
                case 4: display();
                        break;
                case 5: break;
                default: printf("\n Wrong Input");
                }

        }while(ch!=5);

}

void output_deque(){
        int ch;
```

```
        printf("\n Output Restricted Queue");
        do{
                printf("\n 1. Insert at Right\t 2. Insert at Left\t 3. Delete at Left\t 4. Display\t
5. Exit");
                printf("\n Your choice:");
                scanf("%d", &ch);
                switch(ch){
                case 1: in_R();
                        break;
                case 2: in_L();
                        break;
                case 3: de_L();
                        break;
                case 4: display();
                        break;
                case 5: break;
                default: printf("\n Wrong Input");
                }

        }while(ch!=5);
}


int main(){
        int choice;
        printf("\n Menu:-\n 1. Input Restricted Queue\n 2. Output Restricted Queue ");
        printf("\n Your choice:");
        scanf("%d", &choice);
        switch(choice){
                case 1: input_deque();
                        break;
```

```
                case 2: output_deque();
                        break;
                default: break;
        }
        return 0;
}
```

Ouput: