Name: Dhruv Rupareliya (DSY)

Roll No: 71

Experiment no: 4

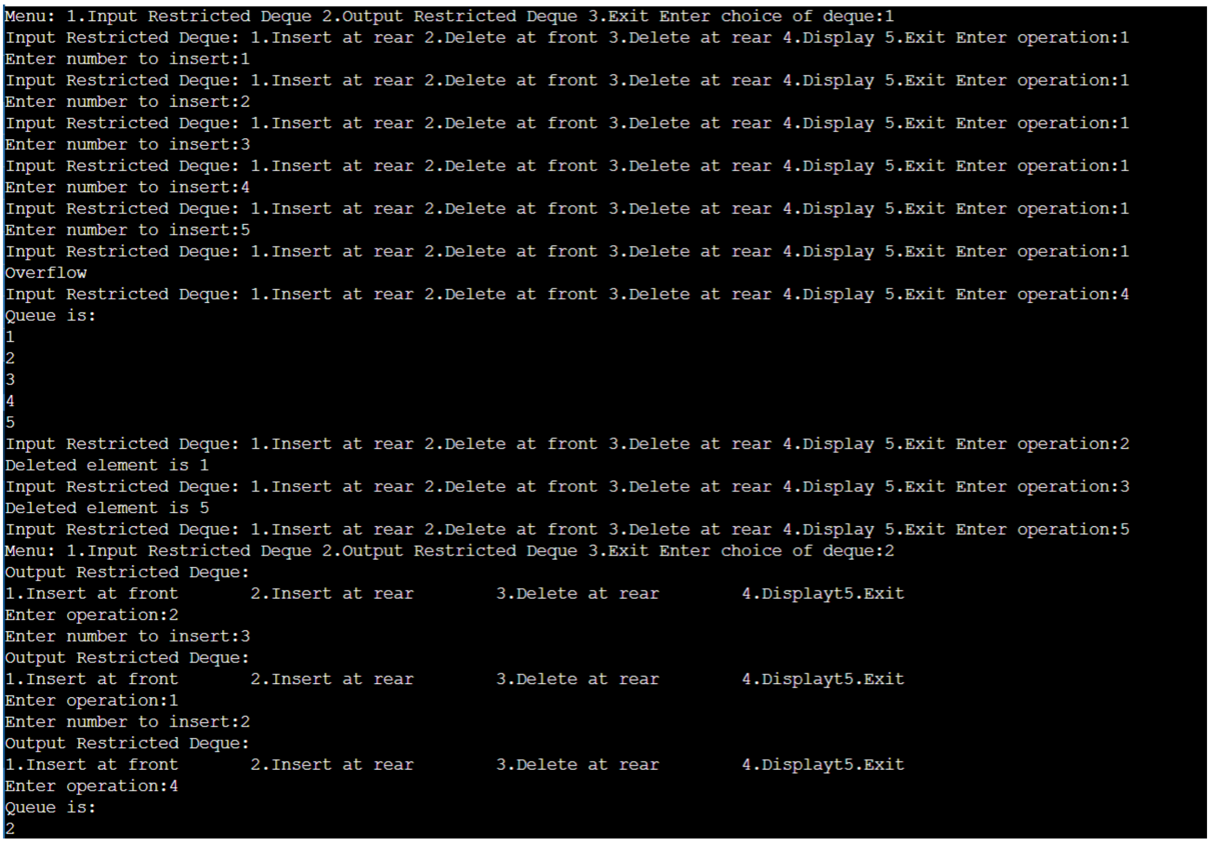
Code:

```
#include<stdio.h>
#define SIZE 5
int q[SIZE], front = -1, rear = -1, c, n;
void input_dq();
void output_dq();
void insertf();
void insertr();
void deletef();
void deleter();
void display();
int main() {
 do {
 printf("Menu: 1.Input Restricted Deque 2.Output Restricted Deque 3.Exit Enter choice of deque:");
 scanf("%d", &c);
 switch (c) {
 case 1:
 input_dq();
 break;
 case 2:
 output_dq();
 break;
 case 3:
 return 0;
 default:
 printf("Invalid Choice\n");
 }
 } while (c != 3);
 return 0;
}
void input_dq() {
 do {
 printf("Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit
Enter operation:");
 scanf("%d", &c);
 switch (c) {
 case 1:
 insertr();
 break;
 case 2:
 deletef();
 break;
 case 3:
 deleter();
 break;
 case 4:
 display();
 break;
 case 5:
 return;
 default:
 printf("Invalid operation\n");
 }
 } while (c != 5);
}
void output_dq() {
 do {
 printf("Output Restricted Deque:\n1.Insert at front\t2.Insert at rear\t3.Delete at rear\t4.Display\
t5.Exit\nEnter operation:");
 scanf("%d", &c);
 switch (c) {
 case 1:
 insertf();
 break;
 case 2:
 insertr();
 break;
 case 3:
 deleter();
 break;
 case 4:
 display();
 break;
 case 5:
 return;
 default:
 printf("Invalid operation\n");
 }
 } while (c != 5);
}
void insertr() {
 if ((front == 0 && rear == SIZE - 1) || (front == rear + 1)) {
 printf("Overflow\n");
 } else {
 printf("Enter number to insert:");
 scanf("%d", &n);
 if (front == -1) {
 front = 0;
 rear = 0;
 } else {
 if (rear == SIZE - 1) {
 rear = 0;
 } else {
 rear++;
 }
 }
 q[rear] = n;
 }
}
void insertf() {
 if ((front == 0 && rear == SIZE - 1) || (front == rear + 1)) {
 printf("Overflow\n");
 } else {
 printf("Enter number to insert:");
 scanf("%d", &n);
 if (front == -1) {
 front = 0;
 rear = 0;
 } else {
 if (front == 0) {
 front = SIZE - 1;
 } else {
 front--;
 }
 }
 q[front] = n;
 }
}
void deleter() {
 if (rear == -1) {
```

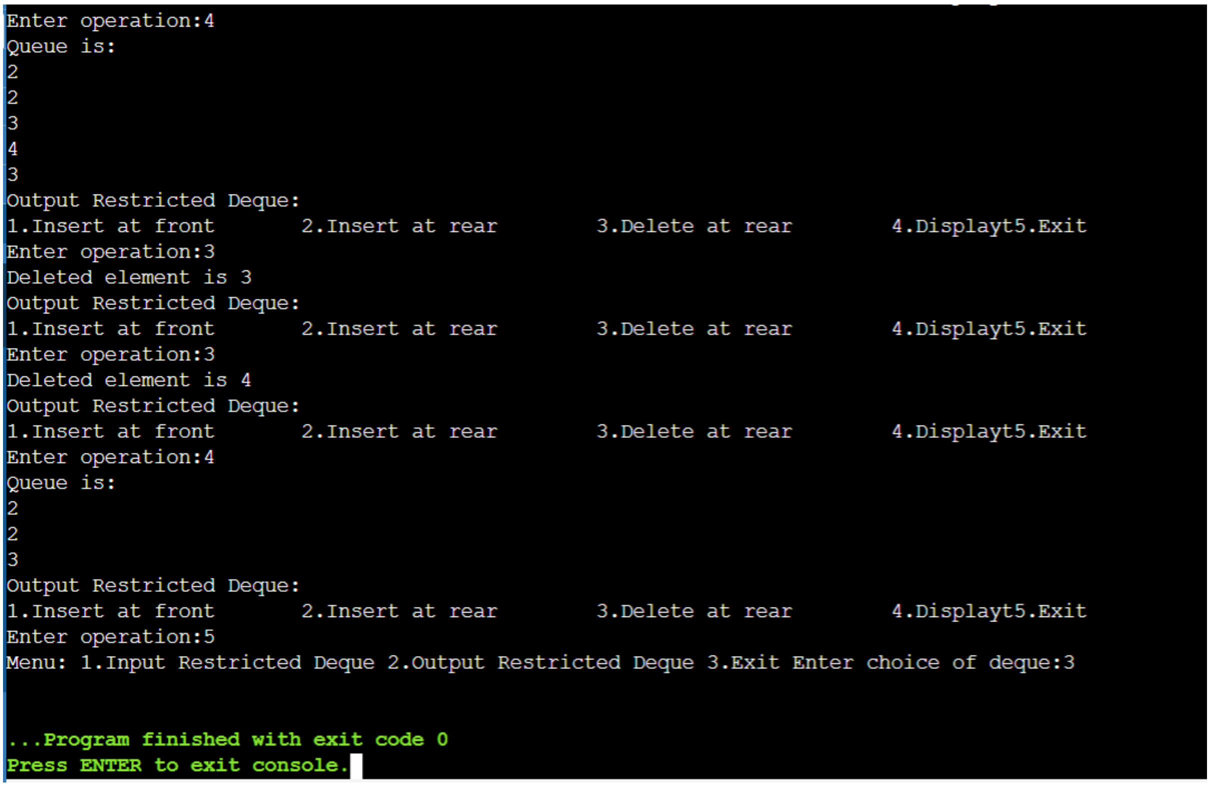
```
 printf("Underflow\n");
 } else {
 printf("Deleted element is %d\n", q[rear]);
 if (rear == 0) {
 rear = SIZE - 1;
 } else {
 rear--;
 }
 }
}
void deletef() {
 if (front == -1) {
 printf("Underflow\n");
 } else {
 printf("Deleted element is %d\n", q[front]);
 if (front == SIZE - 1) {
 front = 0;
 } else {
 front++;
 }
 }
}
void display() {
 int i, j;
 i = front;
 j = rear;
 if (i == -1) {
 printf("Queue is Empty\n");
 } else {
 printf("Queue is:\n");
 if (i <= j) {
```

```
while (i <= j) {
printf("%d\n", q[i]);
i++;
}
} else {
while (i <= SIZE - 1) {
printf("%d\n", q[i]);
i++;
}
for (i = 0; i <= j; i++) {
printf("%d\n", q[i]);
}
}
}
}
```

Output:



```
Menu: 1.Input Restricted Deque 2.Output Restricted Deque 3.Exit Enter choice of deque:1
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Enter number to insert:1
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Enter number to insert:2
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Enter number to insert:3
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Enter number to insert:4
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Enter number to insert:5
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:1
Overflow
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:4
Queue is:
1
2
3
4
5
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:2
Deleted element is 1
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:3
Deleted element is 5
Input Restricted Deque: 1.Insert at rear 2.Delete at front 3.Delete at rear 4.Display 5.Exit Enter operation:5
Menu: 1.Input Restricted Deque 2.Output Restricted Deque 3.Exit Enter choice of deque:2
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:2
Enter number to insert:3
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:1
Enter number to insert:2
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:4
Queue is:
2
```



```
Enter operation:4
Queue is:
2
2
3
4
3
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:3
Deleted element is 3
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:3
Deleted element is 4
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:4
Queue is:
2
2
3
Output Restricted Deque:
1.Insert at front       2.Insert at rear        3.Delete at rear        4.Displayt5.Exit
Enter operation:5
Menu: 1.Input Restricted Deque 2.Output Restricted Deque 3.Exit Enter choice of deque:3


...Program finished with exit code 0
Press ENTER to exit console.
```