

DATA STRUCTURES AND ALGORITHMS LAB ASSIGNMENT – 5

Student Details

- Name: Pratyush Deo Singh
 - Registration Number: 25BCE5101
 - Course / Branch: B. Tech CSE (Core)
 - Semester: 2nd
 - Subject: Data Structures and Algorithms
 - Faculty Name: Dr. Malini A
-

Program 1: Linear Queue (Menu Driven)

Program Code

```
C 1_linear_queue.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int q[MAX];
5  int front = -1, rear = -1;
6
7  void enqueue() {
8      int x;
9      if (rear == MAX - 1) {
10         printf("Overflow\n");
11         return;
12     }
13     if (front == -1)
14         front = 0;
15     scanf("%d", &x);
16     rear++;
17     q[rear] = x;
18 }
19
20 void dequeue() {
21     if (front == -1 || front > rear) {
22         printf("Underflow\n");
23         return;
24     }
25     printf("Deleted %d\n", q[front]);
26     front++;
27 }
28
29 void display() {
30     if (front == -1 || front > rear) {
31         printf("Empty\n");
32         return;
33     }
34     for (int i = front; i <= rear; i++)
35         printf("%d ", q[i]);
36     printf("\n");
37 }
38
39 int main() {
40     int ch;
41     while (1) {
42         scanf("%d", &ch);
43         if (ch == 1) enqueue();
44         else if (ch == 2) dequeue();
45         else if (ch == 3) display();
46         else break;
47     }
48     return 0;
49 }
```

Output

```
1
10
1
20
3
10 20
2
Deleted 10
3
20
4
```

Program 2: Count Elements in Linear Queue

Program Code

```
C 2_count_linear_queue.c > ...
1  #include <stdio.h>
2
3  int main() {
4      int front, rear;
5      scanf("%d %d", &front, &rear);
6
7      if (front == -1 || front > rear)
8          printf("Count = 0");
9      else
10         printf("Count = %d", rear - front + 1);
11
12     return 0;
13 }
14
```

Output

```
2 6
Count = 5
```

Program 3: Reverse Linear Queue

Program Code

```
C 3_reverse_linear_queue.c > ...
1  #include <stdio.h>
2
3  int main() {
4      int n, a[50];
5      scanf("%d", &n);
6
7      for (int i = 0; i < n; i++)
8          scanf("%d", &a[i]);
9
10     for (int i = 0; i < n / 2; i++) {
11         int t = a[i];
12         a[i] = a[n - i - 1];
13         a[n - i - 1] = t;
14     }
15
16     for (int i = 0; i < n; i++)
17         printf("%d ", a[i]);
18
19     return 0;
20 }
21
```

Output

```
5
1 2 3 4 5
5 4 3 2 1
```

Program 4: Circular Queue (Menu Driven)

Program Code

```
C 4_circular_queue.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int q[MAX];
5  int front = -1, rear = -1;
6
7  void enqueue() {
8      int x;
9      if ((rear + 1) % MAX == front) {
10         printf("Overflow\n");
11         return;
12     }
13     scanf("%d", &x);
14     if (front == -1) {
15         front = 0;
16         rear = 0;
17     } else {
18         rear = (rear + 1) % MAX;
19     }
20     q[rear] = x;
21 }
22
23 void dequeue() {
24     if (front == -1) {
25         printf("Underflow\n");
26         return;
27     }
28     printf("Deleted %d\n", q[front]);
29
30     if (front == rear) {
31         front = -1;
32         rear = -1;
33     } else {
34         front = (front + 1) % MAX;
35     }
36 }
37
38 void display() {
39     if (front == -1) {
40         printf("Empty\n");
41         return;
42     }
43     int i = front;
44     while (1) {
45         printf("%d ", q[i]);
46         if (i == rear)
47             break;
48         i = (i + 1) % MAX;
49     }
50     printf("\n");
51 }
52
53 int main() {
54     int ch;
55     while (1) {
56         scanf("%d", &ch);
57         if (ch == 1) enqueue();
58         else if (ch == 2) dequeue();
59         else if (ch == 3) display();
60         else break;
61     }
62     return 0;
63 }
```

Output

```
1
10
1
20
1
30
3
3
10 20 30
2
Deleted 10
3
20 30
4
```

Program 5: Count Elements in Circular Queue

Program Code

```
C 5_count_circular_queue.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int main() {
5      int front, rear;
6      scanf("%d %d", &front, &rear);
7
8      if (front == -1)
9          printf("Count = 0");
10     else if (rear >= front)
11         printf("Count = %d", rear - front + 1);
12     else
13         printf("Count = %d", MAX - front + rear + 1);
14
15     return 0;
16 }
17
```

Output

```
7 2
Count = 6
```

Program 6: Reverse Circular Queue

Program Code

```
C 6_reverse_circular_queue.c > ...
1  #include <stdio.h>
2
3  int main() {
4      int n, a[50];
5      scanf("%d", &n);
6
7      for (int i = 0; i < n; i++)
8          scanf("%d", &a[i]);
9
10     for (int i = 0; i < n / 2; i++) {
11         int t = a[i];
12         a[i] = a[n - i - 1];
13         a[n - i - 1] = t;
14     }
15
16     for (int i = 0; i < n; i++)
17         printf("%d ", a[i]);
18
19     return 0;
20 }
21
```

Ln 21, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32

Output

```
4
11 22 33 44
44 33 22 11
```

Program 7: Deque using Array (Menu Driven)

Program Code

```
C 7_deque.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int dq[MAX];
5  int front = -1, rear = -1;
6
7  int isFull() {
8      return (front == 0 && rear == MAX-1) || (front == rear + 1);
9  }
10
11 int isEmpty() {
12     return front == -1;
13 }
14
15 void insertFront() {
16     int x;
17     if (isFull()) {
18         printf("Overflow\n");
19         return;
20     }
21     scanf("%d", &x);
22
23     if (front == -1)
24         front = rear = 0;
25     else if (front == 0)
26         front = MAX - 1;
27     else
28         front--;
29
30     dq[front] = x;
31 }
32
33 void insertRear() {
34     int x;
35     if (isFull()) {
36         printf("Overflow\n");
37         return;
38     }
39     scanf("%d", &x);
40
41     if (rear == -1)
42         front = rear = 0;
43     else if (rear == MAX - 1)
44         rear = 0;
45     else
46         rear++;
47
48     dq[rear] = x;
49 }
50
51 void deleteFront() {
52     if (isEmpty()) {
53         printf("Underflow\n");
54         return;
55     }
56     printf("Deleted %d\n", dq[front]);
57
58     if (front == rear)
59         front = rear = -1;
60     else if (front == MAX - 1)
61         front = 0;
62     else
63         front++;
64 }
65
66 void deleteRear() {
67     if (isEmpty()) {
68         printf("Underflow\n");
69         return;
70     }
71     printf("Deleted %d\n", dq[rear]);
72 }
```



```
C 7_deque.c > ...
66 void deleteRear() {
67     if (isEmpty()) {
68
69     }
71     printf("Deleted %d\n", dq[rear]);
72
73     if (front == rear)
74         front = rear = -1;
75     else if (rear == 0)
76         rear = MAX - 1;
77     else
78         rear--;
79 }
80
81 void peekFront() {
82     if (isEmpty())
83         printf("Empty\n");
84     else
85         printf("%d\n", dq[front]);
86 }
87
88 void peekRear() {
89     if (isEmpty())
90         printf("Empty\n");
91     else
92         printf("%d\n", dq[rear]);
93 }
94
95 void display() {
96     if (isEmpty()) {
97         printf("Empty\n");
98         return;
99     }
100     int i = front;
101     while (1) {
102         printf("%d ", dq[i]);
103         if (i == rear)
104             break;
105         i = (i + 1) % MAX;
106     }
107     printf("\n");
108 }
109
110 int main() {
111     int ch;
112     while (1) {
113         scanf("%d", &ch);
114         if (ch == 1) insertFront();
115         else if (ch == 2) insertRear();
116         else if (ch == 3) deleteFront();
117         else if (ch == 4) deleteRear();
118         else if (ch == 5) peekFront();
119         else if (ch == 6) peekRear();
120         else if (ch == 7) display();
121         else break;
122     }
123     return 0;
124 }
125
```

Output

```
1
10
2
20
7
10 20
5
10
6
20
3
Deleted 10
7
20
8
```

Program 8: Input Restricted Deque

Program Code

```
C 8_input_restricted_deque.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int dq[MAX];
5  int front=-1, rear=-1;
6
7  int isFull() {
8      return (front==0 && rear==MAX-1) || (front==rear+1);
9  }
10
11 int isEmpty() {
12     return front==0;
13 }
14
15 void insertRear() {
16     int x;
17     if(isFull()) return;
18     scanf("%d",&x);
19     if(front==0) front=rear=0;
20     else if(rear==MAX-1) rear=0;
21     else rear++;
22     dq[rear]=x;
23 }
24
25 void deleteFront() {
26     if(isEmpty()) return;
27     if(front==rear) front=rear=-1;
28     else if(front==MAX-1) front=0;
29     else front++;
30 }
31
32 void deleteRear() {
33     if(isEmpty()) return;
34     if(front==rear) front=rear=-1;
35     else if(rear==0) rear=MAX-1;
36     else rear--;
37 }
38
39 int main() {
40     int ch;
41     while(1){
42         scanf("%d",&ch);
43         if(ch==1) insertRear();
44         else if(ch==2) deleteFront();
45         else if(ch==3) deleteRear();
46         else break;
47     }
48     return 0;
49 }
50
```

Output

```
1
10
1
20
2
3
4
```

Program 9: Output Restricted Deque

Program Code

```
C 9_output_restricted_deque.c > ...
1  #include <stdio.h>
2  #define MAX 10
3
4  int dq[MAX];
5  int front=-1, rear=-1;
6
7  int isFull() {
8      return (front==0 && rear==MAX-1) || (front==rear+1);
9  }
10
11 int isEmpty() {
12     return front==-1;
13 }
14
15 void insertFront(){
16     int x;
17     if(isFull()) return;
18     scanf("%d",&x);
19     if(front==-1) front=rear=0;
20     else if(front==0) front=MAX-1;
21     else front--;
22     dq[front]=x;
23 }
24
25 void insertRear(){
26     int x;
27     if(isFull()) return;
28     scanf("%d",&x);
29     if(front==-1) front=rear=0;
30     else if(rear==MAX-1) rear=0;
31     else rear++;
32     dq[rear]=x;
33 }
34
35 void deleteFront(){
36     if(isEmpty()) return;
37     if(front==rear) front=rear=-1;
38     else if(front==MAX-1) front=0;
39     else front++;
40 }
41
42 int main(){
43     int ch;
44     while(1){
45         scanf("%d",&ch);
46         if(ch==1) insertFront();
47         else if(ch==2) insertRear();
48         else if(ch==3) deleteFront();
49         else break;
50     }
51     return 0;
52 }
53
```

Output

```
1
5
2
15
3
4
```

Program 10: Palindrome Check using Deque

Program Code

```
C 10_palindrome_deque.c > ...
1  #include <stdio.h>
2  #include <string.h>
3  #define MAX 50
4
5  char dq[MAX];
6  int front = -1, rear = -1;
7
8  void insertRear(char c) {
9      if (front == -1)
10         front = rear = 0;
11     else
12         rear++;
13     dq[rear] = c;
14 }
15
16 char deleteFront() {
17     return dq[front++];
18 }
19
20 char deleteRear() {
21     return dq[rear--];
22 }
23
24 int main() {
25     char s[50];
26     scanf("%s", s);
27
28     for (int i = 0; i < strlen(s); i++)
29         insertRear(s[i]);
30
31     while (front < rear) {
32         if (deleteFront() != deleteRear()) {
33             printf("Not Palindrome");
34             return 0;
35         }
36     }
37
38     printf("Palindrome");
39     return 0;
40 }
41
```

Output

```
madam
Palindrome
```

Result

All programs were successfully executed and verified.