DATA STRUCTURES AND ALGORITHMS LAB ASSIGNMENT – 5

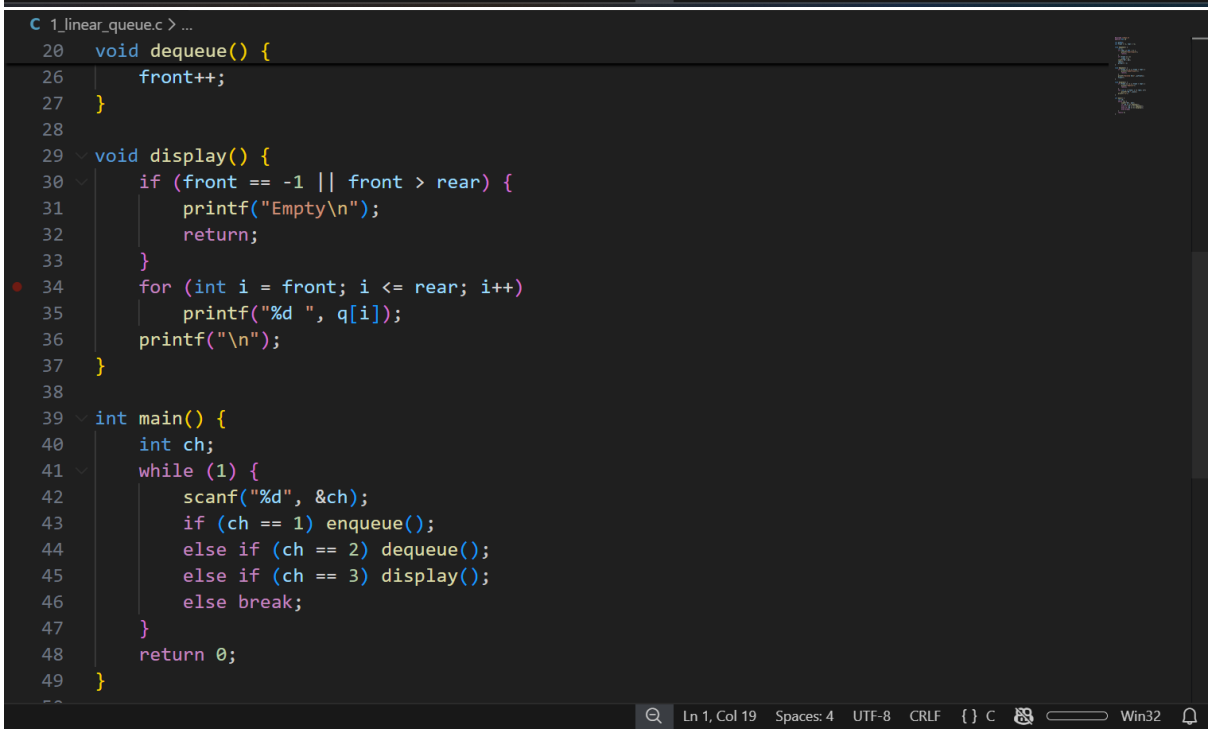
Student Details

- 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
#include <stdio.h>
#define MAX 10

int q[MAX];
int front = -1, rear = -1;

void enqueue() {
    int x;
    if (rear == MAX - 1) {
        printf("Overflow\n");
        return;
    }
    if (front == -1)
        front = 0;
    scanf("%d", &x);
    rear++;
    q[rear] = x;
}

void dequeue() {
    if (front == -1 || front > rear) {
        printf("Underflow\n");
        return;
    }
    printf("Deleted %d\n", q[front]);
```



```c
void dequeue() {
    front++;
}

void display() {
    if (front == -1 || front > rear) {
        printf("Empty\n");
        return;
    }
    for (int i = front; i <= rear; i++)
        printf("%d ", q[i]);
    printf("\n");
}

int main() {
    int ch;
    while (1) {
        scanf("%d", &ch);
        if (ch == 1) enqueue();
        else if (ch == 2) dequeue();
        else if (ch == 3) display();
        else break;
    }
    return 0;
}
```

Output

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

Program 2: Count Elements in Linear Queue

Program Code

2_count_linear_queue.c > ...

```c
#include <stdio.h>

int main() {
    int front, rear;
    scanf("%d %d", &front, &rear);

    if (front == -1 || front > rear)
        printf("Count = 0");
    else
        printf("Count = %d", rear - front + 1);

    return 0;
}

```

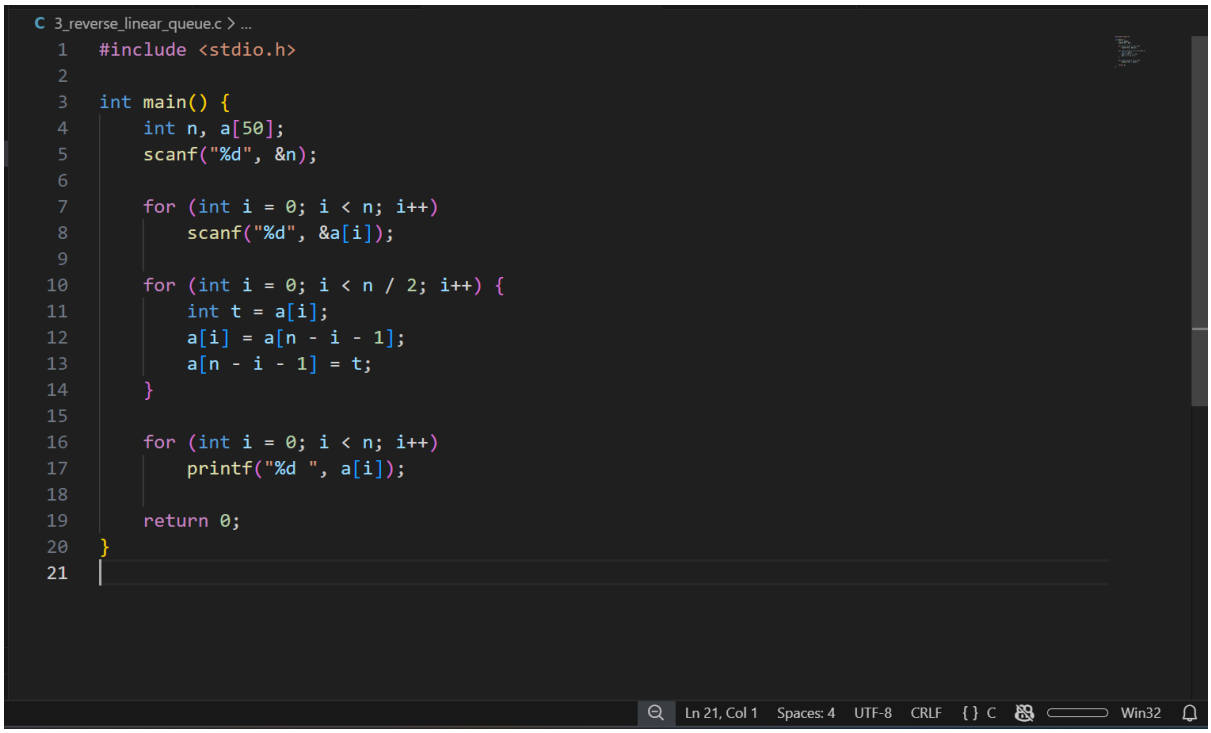
Ln 14, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

Output

```
2 6
Count = 5
```

Program 3: Reverse Linear Queue

Program Code

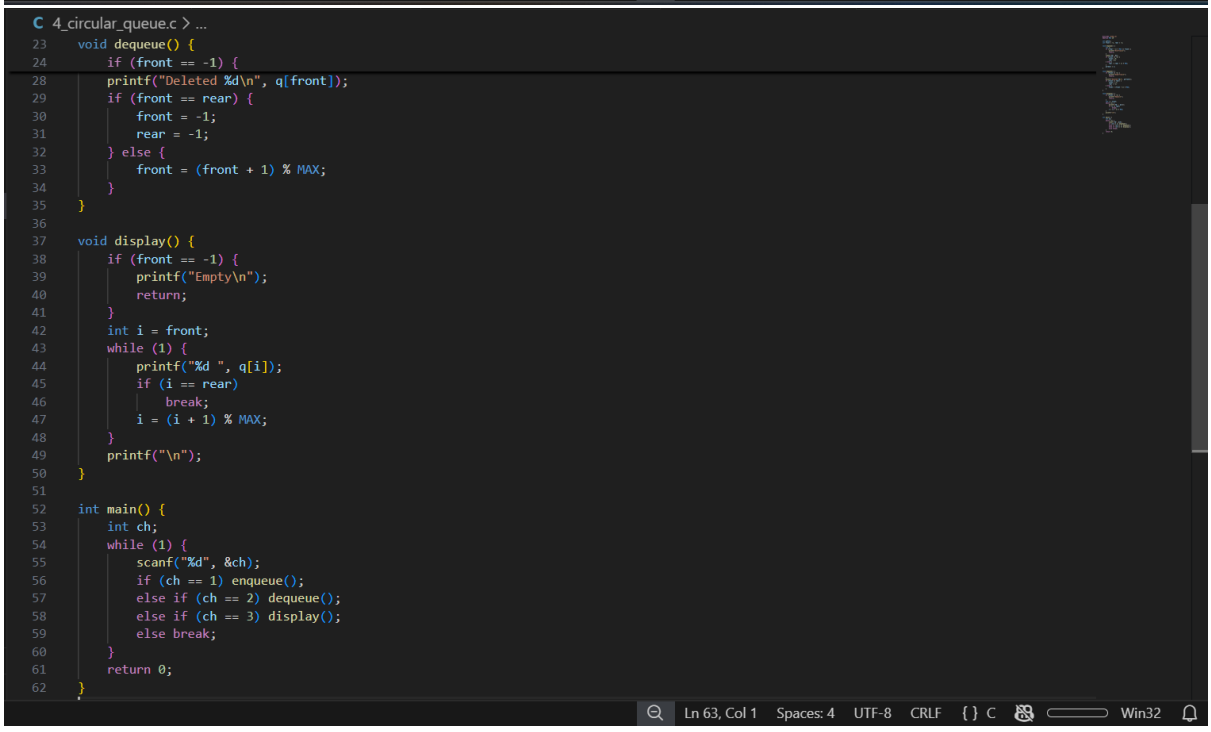


Output

Program 4: Circular Queue (Menu Driven)

Program Code

```c
#include <stdio.h>
#define MAX 10

int q[MAX];
int front = -1, rear = -1;

void enqueue() {
    int x;
    if ((rear + 1) % MAX == front) {
        printf("Overflow\n");
        return;
    }
    scanf("%d", &x);
    if (front == -1) {
        front = 0;
        rear = 0;
    } else {
        rear = (rear + 1) % MAX;
    }
    q[rear] = x;
}

void dequeue() {
    if (front == -1) {
        printf("Underflow\n");
        return;
    }
    printf("Deleted %d\n", q[front]);
```



```c
void dequeue() {
    if (front == -1) {
    printf("Deleted %d\n", q[front]);
    if (front == rear) {
        front = -1;
        rear = -1;
    } else {
        front = (front + 1) % MAX;
    }
}

void display() {
    if (front == -1) {
        printf("Empty\n");
        return;
    }
    int i = front;
    while (1) {
        printf("%d ", q[i]);
        if (i == rear)
            break;
        i = (i + 1) % MAX;
    }
    printf("\n");
}

int main() {
    int ch;
    while (1) {
        scanf("%d", &ch);
        if (ch == 1) enqueue();
        else if (ch == 2) dequeue();
        else if (ch == 3) display();
        else break;
    }
    return 0;
}
```

Output

```
1
10
1
20
1
30
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 > ...

```c
#include <stdio.h>
#define MAX 10

int main() {
    int front, rear;
    scanf("%d %d", &front, &rear);

    if (front == -1)
        printf("Count = 0");
    else if (rear >= front)
        printf("Count = %d", rear - front + 1);
    else
        printf("Count = %d", MAX - front + rear + 1);

    return 0;
}

```

Ln 17, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

Output

```
7 2
Count = 6
```

Program 6: Reverse Circular Queue

Program Code

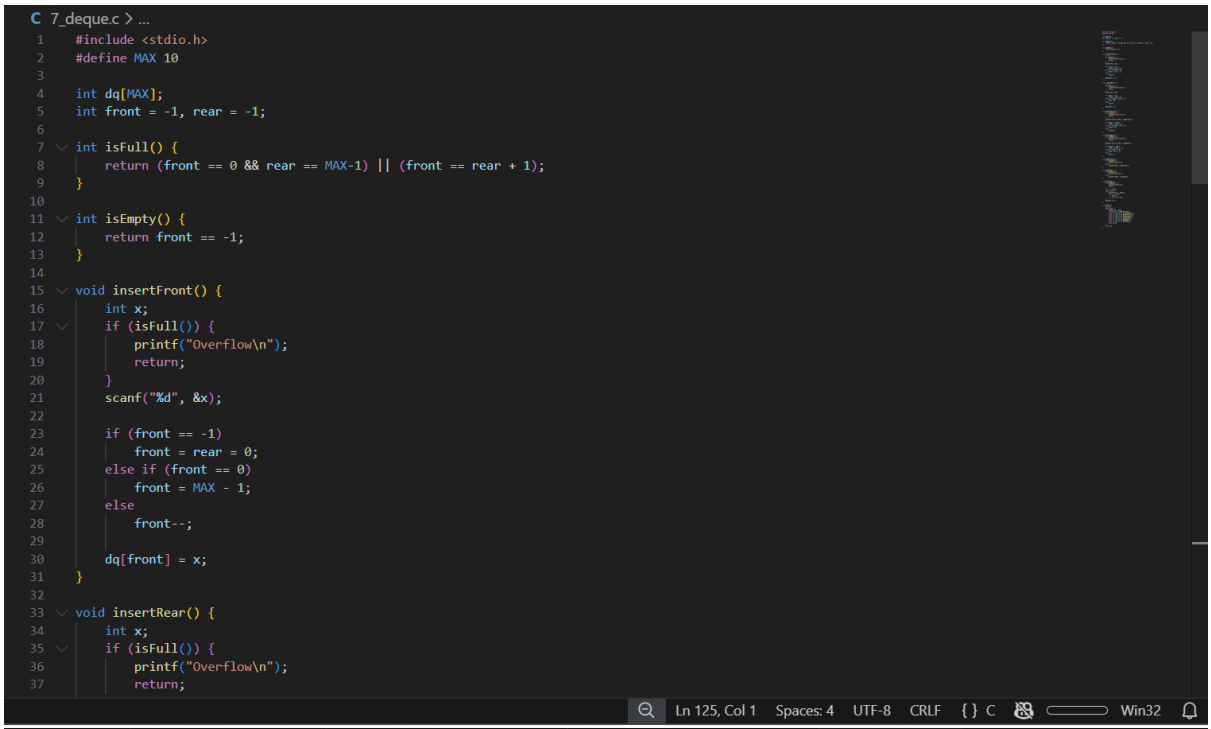
```c
#include <stdio.h>

int main() {
    int n, a[50];
    scanf("%d", &n);

    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);

    for (int i = 0; i < n / 2; i++) {
        int t = a[i];
        a[i] = a[n - i - 1];
        a[n - i - 1] = t;
    }

    for (int i = 0; i < n; i++)
        printf("%d ", a[i]);

    return 0;
}
```

Output

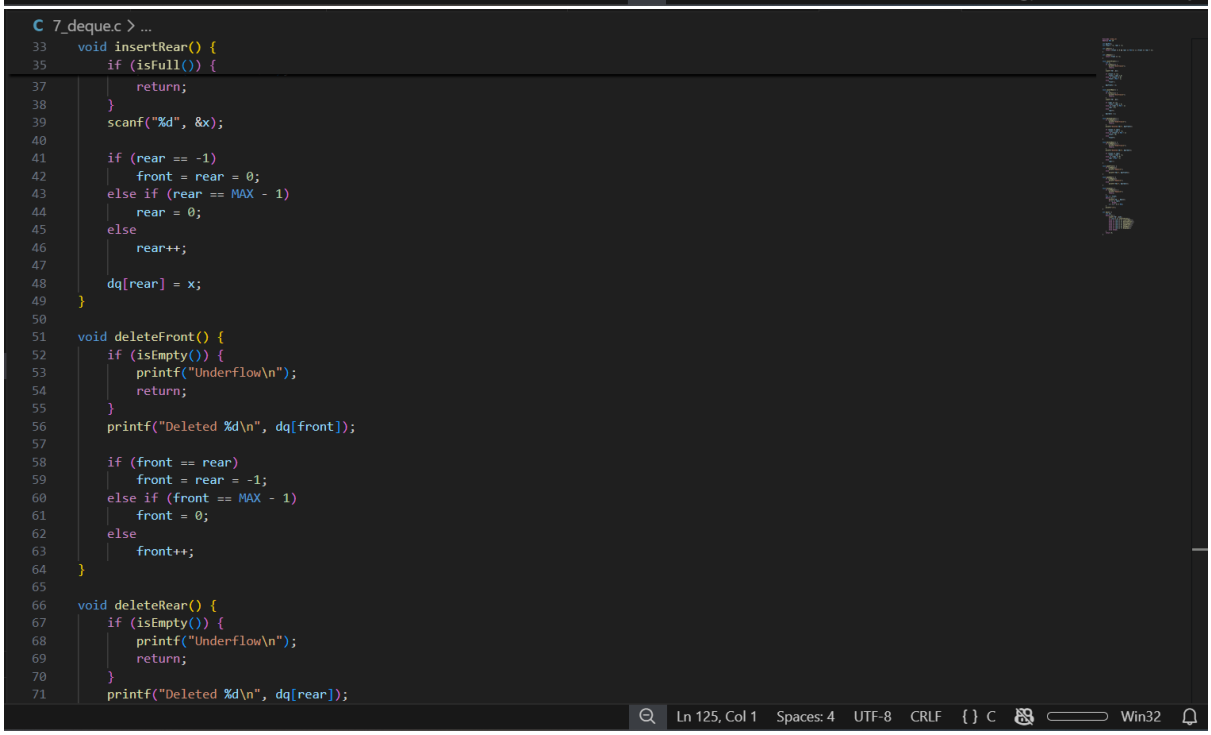
```
4
11 22 33 44
44 33 22 11
```

Program 7: Deque using Array (Menu Driven)

Program Code



```c
#include <stdio.h>
#define MAX 10

int dq[MAX];
int front = -1, rear = -1;

int isFull() {
    return (front == 0 && rear == MAX-1) || (front == rear + 1);
}

int isEmpty() {
    return front == -1;
}

void insertFront() {
    int x;
    if (isFull()) {
        printf("Overflow\n");
        return;
    }
    scanf("%d", &x);

    if (front == -1)
        front = rear = 0;
    else if (front == 0)
        front = MAX - 1;
    else
        front--;

    dq[front] = x;
}

void insertRear() {
    int x;
    if (isFull()) {
        printf("Overflow\n");
        return;
```



```c
void insertRear() {
    if (isFull()) {
        return;
    }
    scanf("%d", &x);

    if (rear == -1)
        front = rear = 0;
    else if (rear == MAX - 1)
        rear = 0;
    else
        rear++;

    dq[rear] = x;
}

void deleteFront() {
    if (isEmpty()) {
        printf("Underflow\n");
        return;
    }
    printf("Deleted %d\n", dq[front]);

    if (front == rear)
        front = rear = -1;
    else if (front == MAX - 1)
        front = 0;
    else
        front++;
}

void deleteRear() {
    if (isEmpty()) {
        printf("Underflow\n");
        return;
    }
    printf("Deleted %d\n", dq[rear]);
```

```c
C 7_deque.c > ...
66  void deleteRear() {
67      if (isEmpty()) {
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;
```

Ln 125, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

```c
C 7_deque.c > ...
95  void display() {
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
```

Ln 125, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

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
#include <stdio.h>
#define MAX 10

int dq[MAX];
int front=-1, rear=-1;

int isFull() {
    return (front==0 && rear==MAX-1) || (front==rear+1);
}

int isEmpty() {
    return front==-1;
}

void insertRear() {
    int x;
    if(isFull()) return;
    scanf("%d",&x);
    if(front==-1) front=rear=0;
    else if(rear==MAX-1) rear=0;
    else rear++;
    dq[rear]=x;
}

void deleteFront() {
    if(isEmpty()) return;
    if(front==rear) front=rear=-1;
    else if(front==MAX-1) front=0;
    else front++;
}

void deleteRear() {
    if(isEmpty()) return;
    if(front==rear) front=rear=-1;
    else if(rear==0) rear=MAX-1;
    else rear--;
}

int main() {
    int ch;
    while(1){
        scanf("%d",&ch);
        if(ch==1) insertRear();
        else if(ch==2) deleteFront();
        else if(ch==3) deleteRear();
        else break;
    }
    return 0;
}
```

Output

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

Program 9: Output Restricted Deque

Program Code

```c
#include <stdio.h>
#define MAX 10

int dq[MAX];
int front=-1, rear=-1;

int isFull() {
    return (front==0 && rear==MAX-1) || (front==rear+1);
}

int isEmpty() {
    return front==-1;
}

void insertFront(){
    int x;
    if(isFull()) return;
    scanf("%d",&x);
    if(front==-1) front=rear=0;
    else if(front==0) front=MAX-1;
    else front--;
    dq[front]=x;
}

void insertRear(){
    int x;
    if(isFull()) return;
    scanf("%d",&x);
    if(front==-1) front=rear=0;
    if(isFull()) return;
    scanf("%d",&x);
    if(front==-1) front=rear=0;
    else if(rear==MAX-1) rear=0;
    else rear++;
    dq[rear]=x;
}

void deleteFront(){
    if(isEmpty()) return;
    if(front==rear) front=rear=-1;
    else if(front==MAX-1) front=0;
    else front++;
}

int main(){
    int ch;
    while(1){
        scanf("%d",&ch);
        if(ch==1) insertFront();
        else if(ch==2) insertRear();
        else if(ch==3) deleteFront();
        else break;
    }
    return 0;
}
```

Output

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

Program 10: Palindrome Check using Deque

Program Code

```c
#include <stdio.h>
#include <string.h>
#define MAX 50

char dq[MAX];
int front = -1, rear = -1;

void insertRear(char c) {
    if (front == -1)
        front = rear = 0;
    else
        rear++;
    dq[rear] = c;
}

char deleteFront() {
    return dq[front++];
}

char deleteRear() {
    return dq[rear--];
}

int main() {
    char s[50];
    scanf("%s", s);

    for (int i = 0; i < strlen(s); i++)
        insertRear(s[i]);
```

Ln 41, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

```c
char deleteFront() {

char deleteRear() {
    return dq[rear--];
}

int main() {
    char s[50];
    scanf("%s", s);

    for (int i = 0; i < strlen(s); i++)
        insertRear(s[i]);

    while (front < rear) {
        if (deleteFront() != deleteRear()) {
            printf("Not Palindrome");
            return 0;
        }
    }

    printf("Palindrome");
    return 0;
}
```

Ln 41, Col 1 Spaces: 4 UTF-8 CRLF { } C Win32

Output

```
madam
Palindrome
```

Result

All programs were successfully executed and verified.