

Name-Jeevan R

Section-I

DSA: Lab Program-3

a) WAP to simulate the working of Linear Queue using an array with the following operations: Insert, Delete and Display, also should print appropriate message for queue empty and overflow conditions.

```
#include <stdio.h>
```

```
#define SIZE 3
```

```
int q[SIZE], front=-1, rear=-1;
```

```
void insert() {
```

```
    if (rear == SIZE-1) {  
        printf("Queue Overflow!\n");  
        return;
```

```
    }
```

```
    int x;
```

```
    printf("Enter value: ");
```

```
    scanf("%d",&x);
```

```
    if (front == -1) front = 0;
```

```
    q[++rear] = x;
```

```
    printf("%d inserted.\n", x);
```

```
}
```

```
void delete() {
```

```
    if (front == -1 || front>rear) {  
        printf("Queue Underflow!\n");  
        return;
```

```
    }
```

```

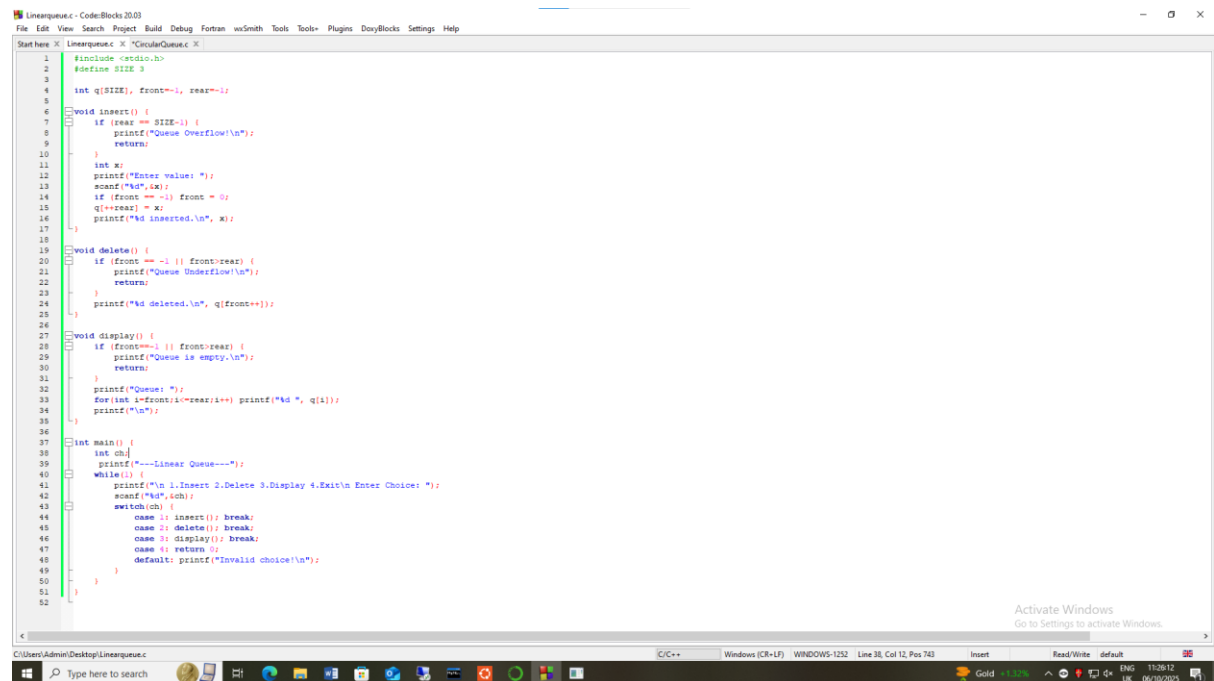
    printf("%d deleted.\n", q[front++]);
}

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

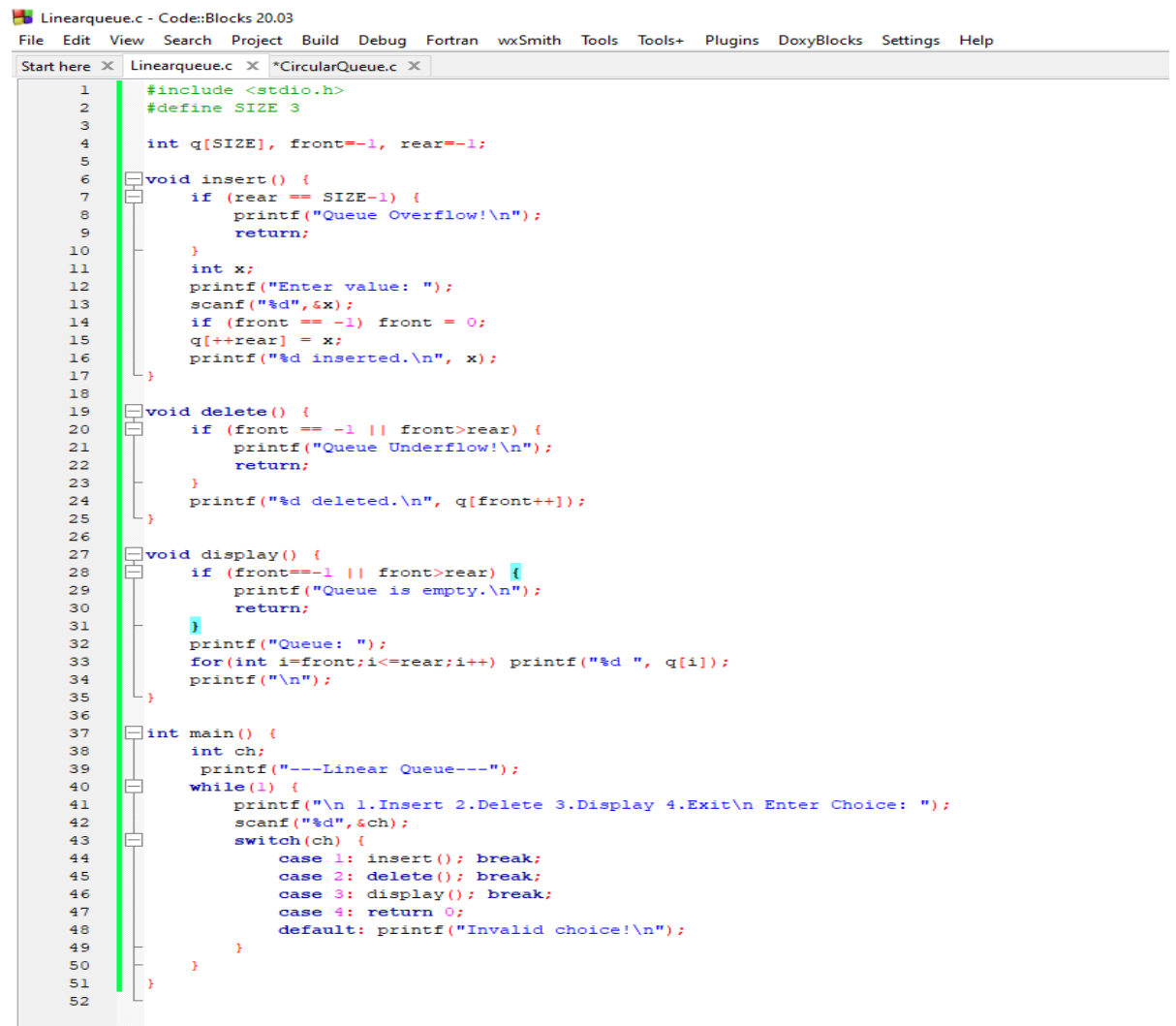
int main() {
    int ch;
    printf("---Linear Queue---");
    while(1) {
        printf("\n 1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
        scanf("%d",&ch);
        switch(ch) {
            case 1: insert(); break;
            case 2: delete(); break;
            case 3: display(); break;
            case 4: return 0;
            default: printf("Invalid choice!\n");
        }
    }
}

```

Code and Expected Output:



```
1 #include <stdio.h>
2 #define SIZE 3
3
4 int q[SIZE], front=-1, rear=-1;
5
6 void insert() {
7     if (rear == SIZE-1) {
8         printf("Queue Overflow!\n");
9         return;
10    }
11    int x;
12    printf("Enter value: ");
13    scanf("%d",&x);
14    if (front == -1) front = 0;
15    q[++rear] = x;
16    printf("%d inserted.\n", x);
17 }
18
19 void delete() {
20     if (front == -1 || front>rear) {
21         printf("Queue Underflow!\n");
22         return;
23     }
24     printf("%d deleted.\n", q[front++]);
25 }
26
27 void display() {
28     if (front== -1 || front>rear) {
29         printf("Queue is empty.\n");
30         return;
31     }
32     printf("Queue: ");
33     for(int i=front;i<=rear;i++) printf("%d ", q[i]);
34     printf("\n");
35 }
36
37 int main() {
38     int ch;
39     printf("----Linear Queue----");
40     while(1) {
41         printf("\n 1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
42         scanf("%d",&ch);
43         switch(ch) {
44             case 1: insert(); break;
45             case 2: delete(); break;
46             case 3: display(); break;
47             case 4: return 0;
48             default: printf("Invalid choice!\n");
49         }
50     }
51 }
52
```



```
1 #include <stdio.h>
2 #define SIZE 3
3
4 int q[SIZE], front=-1, rear=-1;
5
6 void insert() {
7     if (rear == SIZE-1) {
8         printf("Queue Overflow!\n");
9         return;
10    }
11    int x;
12    printf("Enter value: ");
13    scanf("%d",&x);
14    if (front == -1) front = 0;
15    q[++rear] = x;
16    printf("%d inserted.\n", x);
17 }
18
19 void delete() {
20     if (front == -1 || front>rear) {
21         printf("Queue Underflow!\n");
22         return;
23     }
24     printf("%d deleted.\n", q[front++]);
25 }
26
27 void display() {
28     if (front== -1 || front>rear) {
29         printf("Queue is empty.\n");
30         return;
31     }
32     printf("Queue: ");
33     for(int i=front;i<=rear;i++) printf("%d ", q[i]);
34     printf("\n");
35 }
36
37 int main() {
38     int ch;
39     printf("----Linear Queue----");
40     while(1) {
41         printf("\n 1.Insert 2.Delete 3.Display 4.Exit\n Enter Choice: ");
42         scanf("%d",&ch);
43         switch(ch) {
44             case 1: insert(); break;
45             case 2: delete(); break;
46             case 3: display(); break;
47             case 4: return 0;
48             default: printf("Invalid choice!\n");
49         }
50     }
51 }
52
```

C:\Users\Admin\Desktop\Linearqueue.exe

---Linear Queue---

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 3

Queue is empty.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 1

Enter value: 10

10 inserted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 1

Enter value: 20

20 inserted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 1

Enter value: 30

30 inserted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 1

Queue Overflow!

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 3

Queue: 10 20 30

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 2

10 deleted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 3

Queue: 20 30

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 2

20 deleted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 2

30 deleted.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 2

Queue Underflow!

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 3

Queue is empty.

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 1

Queue Overflow!

1.Insert 2.Delete 3.Display 4.Exit

Enter Choice: 4

Process returned 0 (0x0) execution time : 64.044 s

Press any key to continue.