S.No: 5	Exp. Name: ***Operations on Queue using Arrays***	**Date:2025-08-17**

Aim:

Write a C program to implement queue using arrays.

Input Format:

The program will repeatedly prompt the user with a menu to perform operations on the queue.

1. **Enqueue:** The user will be prompted to enter an integer value to be added to the queue.
2. **Dequeue:** No additional input is needed.
3. **Display:** No additional input is needed.
4. **Is Empty:** No additional input is needed.
5. **Size:** No additional input is needed.
6. **Exit:** No additional input is needed.

Output Format:

1. **Enqueue:**
   If the queue is not full, the output will be: "Successfully inserted." If the queue is full, the output will be: "Queue is overflow."
2. **Dequeue:**
   If the queue is not empty, the output will be: "Deleted element = <X>" where X is the element removed from the queue.If the queue is empty, the output will be: "Queue is underflow."
3. **Display:**
   If the queue is empty, the output will be: "Queue is empty." Otherwise, the output will list the elements in the queue in the order from front to rear, e.g., "Elements in the queue : 10 20"
4. **Is Empty:**
   If the queue is empty, the output will be: "Queue is empty." Otherwise, the output will be: "Queue is not empty."
5. **Size:**
   If the queue is empty, the output will be: "Queue size : 0**"** Otherwise, the output will display the current number of elements, e.g., "Queue size : 3"
6. **Exit:**
7. The program terminates without any additional output.

Note:

- The driver code is provided; you need to implement the logic in the file **QueueOperations.c**.
- Refer to visible test cases to strictly match with input/output layout.

Source Code:

QueueUsingArray.c

```
#include <stdlib.h>
#include <stdio.h>
#include "QueueOperations.c"
int main() {
   int op, x;
   while(1) {
      printf("1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit\n");
      printf("Enter your option : ");
      scanf("%d",&op);
      switch(op) {
         case 1:
            printf("Enter element : ");
            scanf("%d",&x);
            enqueue(x);
            break;
         case 2:
            dequeue();
            break;
         case 3:
            display();
            break;
         case 4:
            isEmpty();
            break;
         case 5:
            size();
            break;
         case 6: exit(0);
      }
   }
}
```

QueueOperations.c

```
//define a variable MAX to 10
#define MAX 10
//declare the size of the array to be MAX
int queue[MAX];
//define the front and rear to -1
int front = -1, rear = -1;
void enqueue(int x){
   // write your code here to enqueue an element
   if(rear == MAX){
      printf("Queue is overflow.\n");
      return;
   }
   if(front==-1) front=0;
   queue[++rear]=x;
   printf("Successfully inserted.\n");
}
void dequeue() {
   // write your code here to dequeue an element
   if(front==-1 || front>rear){
      printf("Queue is underflow.\n");
      return;
   }
   printf("Deleted element = %d\n", queue[front++]);
}
void display() {
   // write your code here to display the queue
   if(front == -1 || front > rear){
      printf("Queue is empty.\n");
      return;
   }
   printf("Elements in the queue : ");
   for(int i = front; i <= rear; i++){
      printf("%d ", queue[i]);
   }
   printf("\n");
}
void size() {
   // write your code here to get the size of the queue
   int s = 0;
   if(front == -1 || front>rear){
      printf("Queue size : 0\n");
      return;
   }
   for(int i = front; i<=rear; i++){
      s = s + 1;
   }
   printf("Queue size : %d\n", s);
}
void isEmpty() {
   // write your code here to check whether the queue is empty not
   if(front == -1 || front>rear)
      printf("Queue is empty.\n");
   else
      printf("Queue is not empty.\n");
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 2
Enter your option :  2
Queue is underflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 3
Enter your option :  3
Queue is empty. 4
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 4
Enter your option :  4
Queue is empty. 5
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 5
Enter your option :  5
Queue size : 0 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  14
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  78
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  53
Successfully inserted. 3
```

```
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 3
Enter your option :  3
Elements in the queue : 14 78 53  5
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 5
Enter your option :  5
Queue size : 3 6
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 6
Enter your option :  6
```

Test Case - 2

User Output

```
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  25
Successfully inserted. 2
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 2
Enter your option :  2
Deleted element = 25 2
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 2
Enter your option :  2
Queue is underflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 3
Enter your option :  3
Queue is empty. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  65
Successfully inserted. 3
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 3
Enter your option :  3
Elements in the queue : 65  4
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 4
Enter your option :  4
Queue is not empty. 2
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 2
Enter your option :  2
Deleted element = 65 4
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 4
Enter your option :  4
Queue is empty. 5
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 5
Enter your option :  5
Queue size : 0 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  63
Successfully inserted. 5
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 5
Enter your option :  5
Queue size : 1 6
```

```
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 6
Enter your option :  6
```

Test Case - 3

User Output

```
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  10
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  20
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  30
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  40
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  50
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  60
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  70
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  80
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  90
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  100
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
Enter your option :  1
Enter element :  12
Queue is overflow. 1
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 1
```

```
Enter your option :  1
Enter element :  16
Queue is overflow. 6
1.Enqueue 2.Dequeue 3.Display 4.Is Empty 5.Size 6.Exit 6
Enter your option :  6
```