Program

```c
#include <stdio.h>

#define MAX 5

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

void enqueue(int item)
{
    if (rear == MAX - 1)
    {
        printf("Queue overflow!!\n");
    }
    else
    {
        if (front == -1)
        {
            front = 0;
        }
        rear++;
        queue[rear] = item;
        printf("Element enqueued in the queue is %d\n", item);
    }
}

int dequeue()
{
    int item;
    if (front == -1)
    {
        printf("Queue underflow!!\n");
        return -1;
    }
    else
    {
        item = queue[front];
        if (front == rear)
        {
            front = rear = -1;
        }
        else
        {
            front++;
        }
        return item;
    }
}

void display()
{
    int i;
    if (front == -1)
    {
        printf("Queue is empty!!\n");
    }
    else
    {
        printf("Queue elements are:\n");
        for (i = front; i <= rear; i++)
        {
```

Experiment No: 8
Date:

QUEUE IMPLEMENTATION USING ARRAY

Aim:

To implement Queue using array

Algorithm:

1. Start
2. Define MAX as 5
3. Declare queue[MAX]
4. Initialize front = -1 and rear = -1
5. Define function enqueue (item)

```
        if (rear = MAX - 1)
              print "Queue overflow!!"
        else
            if (front = -1)
                front = 0
            endif
            rear++
            queue[rear] = item
            print "Element enqueued in the queue is ",item
        endif

   end of function
```

6. Define function dequeue()

```
        if (front = -1)
            print "Queue underflow!!"
            return -1
        else
            item = queue[front]
            if (front = rear)
                front = rear = -1
            else
                 front++
            endif
           return item
```

```
            printf("%d\n", queue[i]);
        }
    }
}

int main()
{
    int choice, item;
    while (1)
    {
        printf("\n1.Enqueue\n2.Dequeue\n3.Display\n4.Exit");
        printf("\nEnter an option(1-4): ");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            printf("Enter element to enqueue: ");
            scanf("%d", &item);
            enqueue(item);
            break;
        case 2:
            item = dequeue();
            if (item != -1)
            {
                printf("%d is the dequeued element\n", item);
            }
            break;
        case 3:
            display();
            break;
        case 4:
            return 0;
        default:
            printf("Invalid option!!\n");
        }
    }
}
```

Output

```

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 23
Element enqueued in the queue is 23

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 11
Element enqueued in the queue is 11

1.Enqueue
2.Dequeue
3.Display
```

```
        endif

    end of function
```

7. Define function display()

```
        if (front = -1)
            print "Queue is empty!!"
        else
           print "Queue elements are: "
           for(int i = front; i <= rear; i++)
                print queue[i]
           endfor
        endif

    end of function
```

8. Create main function

```
    main()
            while(1)
                  printf("1.Enqueue 2.Dequeue 3.Display 4.Exit")
                  Read the user's choice
                  switch(choice)
                        case 1:
                          printf("Enter element to enqueue: ");
                          scanf("%d", &item);
                          enqueue(item);
                        case 2:
                          item = dequeue();
                          printf("%d is the dequeued element", item);
                        case 3:
                          display();
                        case 4:
                          return 0;
                        default:
                          printf("Invalid option!!");
                 endswitch

   end of function
```

9. Stop

```
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 28
Element enqueued in the queue is 28

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 45
Element enqueued in the queue is 45

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 95
Element enqueued in the queue is 95

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 1
Enter element to enqueue: 123
Queue overflow!!
```

```
1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 2
23 is the dequeued element

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 2
11 is the dequeued element

1.Enqueue
2.Dequeue
3.Display
4.Exit
Enter an option(1-4): 3
Queue elements are:
28
45
95
```

Result:

Program has been executed successfully and obtained the output