S.No: 20	Exp. Name: ***Write a C program to implement different Operations on Queue using Dynamic Array***	Date:2023-06-11

Aim:

Write a program to implement queue using **dynamic array**.

In this queue implementation has

1. a pointer 'queue' to a dynamically allocated array (used to hold the contents of the queue)
2. an integer 'maxSize' that holds the size of this array (i.e the maximum number of data that can be held in this array)
3. an integer 'front' which stores the array index of the first element in the queue
4. an integer 'rear' which stores the array index of the last element in the queue.

ID: 224G1A0541
2022-2026-CSE-A
Srinivasa Ramanujan Institute of Technology

```
Sample Input and Output:
        Enter the maximum size of the queue : 3
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 2
        Queue is underflow.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 3
        Queue is empty.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 1
        Enter element : 15
        Successfully inserted.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 1
        Enter element : 16
        Successfully inserted.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 1
        Enter element : 17
        Successfully inserted.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 1
        Enter element : 18
        Queue is overflow.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 3
        Elements in the queue : 15 16 17
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 2
        Deleted element = 15
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 2
        Deleted element = 16
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 3
        Elements in the queue : 17
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 2
        Deleted element = 17
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 3
        Queue is empty.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 2
        Queue is underflow.
        1.Enqueue 2.Dequeue 3.Display 4.Exit
        Enter your option : 4
```

Source Code:

QUsingDynamicArray.c

```
#include<stdio.h>
#include<conio.h>
int *queue;
```

```
int front,rear;
int maxSize;
void initQueue()
{
   queue = (int *)malloc(maxSize*sizeof(int));
   front = -1;
   rear = -1;
}
void enqueue(int x)
{
   if(rear == maxSize - 1)
   {
      printf("Queue is overflow.\n");
   }
   else
   {
      rear++;
      queue[rear] = x;
      printf("Successfully inserted.\n");
   }
   if(front == -1)
   {
      front++;
   }
}
void dequeue()
{
   if(front == -1)
   {
      printf("Queue is underflow.\n");
   }
   else
   {
      printf("Deleted element = %d\n",*(queue+front));
      if(rear == front)
      {
         rear=front=-1;
      }
      else
      {
         front++;
      }
   }
}
void display()
{
   if(front==-1&&rear==-1)
   {
      printf("Queue is empty.\n");
   }
   else
   {
      printf("Elements in the queue : ");
      for(int i=front;i<=rear;i++)
      {
         printf("%d ",*(queue+i));
```

```
      }
      printf("\n");
   }
}
int main()
{
   int op,x;
   printf("Enter the maximum size of the queue : ");
   scanf("%d",&maxSize);
   initQueue();
   while(1)
   {
      printf("1.Enqueue 2.Dequeue 3.Display 4.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:
         exit(0);
      }
   }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the maximum size of the queue : 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Queue is underflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Queue is empty. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 15
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 16
Successfully inserted. 1

1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 17
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 18
Queue is overflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the queue : 15 16 17 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 15 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 16 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the queue : 17 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 17 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Queue is empty. 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Queue is underflow. 4
1.Enqueue 2.Dequeue 3.Display 4.Exit 4
Enter your option : 4

Test Case - 2
User Output
Enter the maximum size of the queue : 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 34
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 56
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 45
Queue is overflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the queue : 34 56 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2

Deleted element = 34 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 56 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Queue is underflow. 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Queue is underflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Queue is empty. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 56
Successfully inserted. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the queue : 56 4
1.Enqueue 2.Dequeue 3.Display 4.Exit 4
Enter your option : 4