S.No: 13	Exp. Name: ***Implementation of Circular Queue using Dynamic Array***	Date: 2022-08-10

Aim:

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

In this `circular 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.

```
Sample Input and Output:
    Enter the maximum size of the circular queue : 3
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 2
    Circular queue is underflow.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 3
    Circular queue is empty.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 1
    Enter element : 111
    Successfully inserted.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 1
    Enter element : 222
    Successfully inserted.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 1
    Enter element : 333
    Successfully inserted.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 1
    Enter element : 444
    Circular queue is overflow.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 3
    Elements in the circular queue : 111 222 333
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 2
    Deleted element = 111
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 1
    Enter element : 444
    Successfully inserted.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 3
    Elements in the circular queue : 222 333 444
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 2
    Deleted element = 222
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 2
    Deleted element = 333
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 2
    Deleted element = 444
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 3
    Circular queue is empty.
    1.Enqueue 2.Dequeue 3.Display 4.Exit
    Enter your option : 4
```

Source Code:

CQueueUsingDynamicArray.c

```
#include <stdio.h>
#include <stdlib.h>
#include "CQueueUsingDynamicArray1.c"

int main() {
   int op, x;
   printf("Enter the maximum size of the circular queue : ");
   scanf("%d", &maxSize);
   initCircularQueue();
   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);
      }
   }
}
```

CQueueUsingDynamicArray1.c

```
#include<stdio.h>
int maxSize;
int *cq;
int front=-1,rear=-1;
void initCircularQueue()
{
   cq = (int *)malloc(sizeof(int)*maxSize);
}
void enqueue(int n)
{
   if(((front==0)&&(rear==maxSize-1))||(front==rear+1))
   {
      printf("Circular queue is overflow.\n");
   }
   else
   {
      if((front==-1)&&(rear==-1))
      {
         front=rear=0;
      }
      else if(rear==maxSize-1)
```

```
      {
         rear=0;
      }
      else
      {
         rear=rear+1;
      }
      cq[rear]=n;
      printf("Successfully inserted.\n");
   }

}
void dequeue()
{
   int value=cq[front];
   if(front==-1&&rear==-1)
   printf("Circular queue is underflow.\n");
   else
   {
      if(front==rear)
      front = rear =-1;
      else if(front == maxSize-1)
      front=0;
      else
      front=front+1;
      printf("Deleted element = %d\n",value);
   }
}
void display()
{
   int i;
   if((front==-1)&&(rear==-1))
   {
      printf("Circular queue is empty.\n");
   }
   else if(front<=rear)
   {
      printf("Elements in the circular queue : ");
      for(i=front;i<=rear;i++)
      {
         printf("%d ",cq[i]);
      }
      printf("\n");
   }
   else
   {
      printf("Elements in the circular queue : ");
      for(i=front;i<=maxSize-1;i++)
      {
         printf("%d ",cq[i]);
      }
      for(i=0;i<=rear;i++)
      {
         printf("%d ",cq[i]);
      }
      printf("\n");
```

```
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1
User Output
Enter the maximum size of the circular queue : 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Circular queue is underflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Circular queue is empty. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 111
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 222
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 333
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 444
Circular queue is overflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 111 222 333 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 111 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 444
Successfully inserted. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 222 333 444 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 222 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 333 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 444 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3

Test Case - 1
Enter your option : 3
Circular queue is empty. 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 circular queue : 4
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Circular queue is empty. 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Circular queue is underflow. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 45
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 99
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 32
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 26
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 37
Circular queue is overflow. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 45 99 32 26 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 45 2
1.Enqueue 2.Dequeue 3.Display 4.Exit 2
Enter your option : 2
Deleted element = 99 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 32 26 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 58

Test Case - 2
Successfully inserted. 1
1.Enqueue 2.Dequeue 3.Display 4.Exit 1
Enter your option : 1
Enter element : 27
Successfully inserted. 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 32 26 58 27 3
1.Enqueue 2.Dequeue 3.Display 4.Exit 3
Enter your option : 3
Elements in the circular queue : 32 26 58 27 4
1.Enqueue 2.Dequeue 3.Display 4.Exit 4
Enter your option : 4

ID: 219X1A04E7

2021-2025-ECE-B

G Pulla Reddy Engineering College (Autonomous)