DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMMING LABORATORY (CSE 351)

ASSIGNMENT 7

Asmit De
10/CSE/53

Date: 03.11.2011

Program 1: Circular Queue

Source Code –

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#define SIZE 10

typedef struct Queue
{
	int _queue[SIZE];
	int front, rear;
}Queue;

void initQueue(Queue *q)
{
	q->front = -1;
	q->rear = -1;
}

int isFull(Queue *q)
{
	if((q->rear == SIZE-1 && q->front == -1) || (q->front - q->rear == 1))
		return 1;
	else
		return 0;
}

int isEmpty(Queue *q)
{
	if(q->front == q->rear)
		return 1;
	else
		return 0;
}

int enqueue(Queue *q, int data)
{
	char ch;
	if(isFull(q))
	{
		printf("\n\aQueue is full...\nDo you want to overwrite the last
entered elemnet? (y/n): ");
		ch = getche();
		if(ch == 'y' || ch == 'Y')
		{
			q->_queue[q->rear] = data;
			return 0;
		}
		else
			return 0;
	}
```

```
	if(q->rear == SIZE-1)
		q->rear = -1;

	q->_queue[++q->rear] = data;
	return 0;
}

int dequeue(Queue *q, int *data)
{
	if(isEmpty(q))
	{
		printf("\n\aError: Queue is empty...");
		return 1;
	}

	*data = q->_queue[++q->front];
	return 0;
}

void displayQueue(Queue *q)
{
	int i;
	printf("Queue: ");
	for(i = q->rear; i != -1; i--)
	{
		printf("%d ", q->_queue[i]);
		if(i == q->front+1)
			break;
	}

	if(i == -1)
	{
		for(i = SIZE-1; i>= q->front+1; i--)
			printf("%d ", q->_queue[i]);
	}
}

int main()
{
	Queue *queue = (Queue*)malloc(sizeof(Queue));
	int *data = (int*)malloc(sizeof(int));
	char choice;

	initQueue(queue);
	while(1)
	{
		clrscr();
		puts("MENU");
		puts("\nKey \tFunction");
		puts("1 \tDisplay Queue");
		puts("2 \tEnqueue Data");
		puts("3 \tDequeue Data");
		puts("4 \tClear Queue");
		puts("X \tExit");
		printf("\nEnter choice...");
		choice = getch();
		fflush(stdin);

		switch(choice)
		{
		case '1':
			clrscr();

			if(isEmpty(queue))
				printf("\n\a:Error: Queue is empty...");
			else
				displayQueue(queue);
```

```
                printf("\n\nPress any key to return to menu...");
                getch();
                break;

        case '2':
                clrscr();

                printf("Enter data: ");
                scanf("%d", data);
                if(!enqueue(queue, *data))
                        printf("\nData queued successfully...");

                printf("\n\nPress any key to return to menu...");
                getch();
                break;

        case '3':
                clrscr();

                if(!dequeue(queue, data))
                        printf("\nData retrieved: %d", *data);

                printf("\n\nPress any key to return to menu...");
                getch();
                break;

        case '4':
                clrscr();

                initQueue(queue);
                printf("Queue cleared successfully...");

                printf("\n\nPress any key to return to menu...");
                getch();
                break;

        case 'X':
        case 'x':
                exit(0);

        default:
                clrscr();

                printf("\aError: Invalid Input...");

                printf("\n\nPress any key to return to menu...");
                getch();
        }
    }
}
```