CS23302 – DATA STRUCTURES LABORATORY

CIRCULARLY SINGLE LINKED LIST, DOUBLY LINKED LIST AND APPLICATIONS

ON THE SPOT

DATE: 6.8.26 **MARKS: 5+ 5**

1. Modify doublylinkedlist.h to include the following functions:
   a. function prints the DL list in reverse - void print_reverse(LIST *head);
   a. function and merge two DL lists to create new DL list such that odd elements of the new list are from list1 and even elements from list2 –
   void merge_odd_even(LIST *head1, LIST *head2, LIST *merged);

2. Implement circular Queue using circularly Single linked List - A queue built on top of the circular singly linked list from linked_list.h / .c.
   Only the REAR of the underlying circular list is stored here. Because the list is circular, the FRONT of the queue is always **rear->next** - no separate front pointer is needed

```
#ifndef CQUEUE_H
#define CQUEUE_H

/* reuses NODE, LIST, element_type, create_node()from linked_list.h */

#include "linked_list.h"
typedef struct {
   LIST rear;   /* NULL when the queue is empty */
} CQueue;

void cqueueInit(CQueue *q);
int  cqueueIsEmpty(CQueue *q);
void cqueueEnqueue(CQueue *q, element_type value);
 /* 1 = success, 0 = empty */
int  cqueueDequeue(CQueue *q, element_type *out);
void cqueuePrint(CQueue *q);
void cqueueFree(CQueue *q);

#endif
```