

Data Structures **Laboratory**

Assignment 1

Name : Katkar Prathamesh Shivaji

Enrollment No. : 18114038

Batch : 02

Branch : CSE

Email ID : prathameshkatkar11@gmail.com

Problem Statement 1:

Write a C program to create a student management system, where the students' information are stored in a doubly circular linked list, as shown in Figure 1. The structure of each node from the list is shown in Figure 2. Initially, the circular doubly linked list is empty and the student personal data is entered from the filename "StudentData.xlsx" that contains the data of 13 students (name, D.O.B., address and phone no) in tabular form. The StudentData.xlsx file can be converted into a CSV file using Libreoffice or into any other file format readable from your C program. The program should have the following operations: insert, delete, search, modify, sort and print. While inserting, a unique roll number in the linked list is assigned to each student, where the starting roll number should be 101 and the list should always be in sorted according to their roll number (ascending order). However, when a deletion operation is performed, the roll number of the deleted student node is stored in a queue named unusedRollNo. These deleted roll numbers from the unusedRollNo queue will be allotted to the new students on next insertion operations.

Perform the testing of your code with the following

TestCases:

(Initially the list is empty)

1. delete (roll number 108) - delete the student node with roll number 108
2. insert - insert first student data from the "StudentData.xlsx" file (Row2)
3. insert - insert second student data from the "StudentData.xlsx" file (Row3)
4. insert - insert 3rd student data from the "StudentData.xlsx" file (Row4)
5. insert - insert 4th student data from the "StudentData.xlsx" file (Row5)
6. delete (roll number 102) - delete the student node with roll number 102
7. delete (roll number 101) - delete the student node with roll number 101

8. insert - insert 5th student data from the "StudentData.xlsx" file (Row6)
 9. insert - insert 6th student data from the "StudentData.xlsx" file (Row7)
 10. insert - insert 7th student data from the "StudentData.xlsx" file (Row8)
 11. print - print the linked list with the roll number, name and D.O.B
 12. sort (name) - sort the name according to student names
 13. print - print the linked list with the roll number, name and D.O.B
 14. modify (roll number 103) - modify the student node having roll number 103
 15. print - print the linked list with all the records for each student
- Note: In 'modify' function, the programmer can update the other fields except the roll number of a student.

Data Structures used:

1. Queue Data structure: Used to hold the discarded roll numbers
2. Circular Doubly Linked List: Used to hold the nodes containing the information.

Algorithms:

1. Quicksort for sorting the array
2. Linear Search for searching an element

Snapshots of code running:
(Using the driver program)

1. delete (roll number 108) - delete the student node with roll number 108

```
psk@predator:~/Desktop/L1_18114038/src/Problem 1$ gcc prob1.c
psk@predator:~/Desktop/L1_18114038/src/Problem 1$ ./a.out
Nothing to delete.
Nothing to print.
```

2. insert - insert first student data from the "StudentData.xlsx" file (Row2)

```
Roll_no = 101
Name = Priyanka Chopra
Date of Birth = 18 Jul 95
```

3. insert - insert second student data from the "StudentData.xlsx" file (Row3)

```
Roll_no = 101  
Name = Priyanka Chopra  
Date of Birth = 18 Jul 95  
Roll_no = 102  
Name = Virat Kohli  
Date of Birth = 5 Nov 97
```

4. insert - insert 3rd student data from the "StudentData.xlsx" file (Row4)

```
Roll_no = 101  
Name = Priyanka Chopra  
Date of Birth = 18 Jul 95  
Roll_no = 102  
Name = Virat Kohli  
Date of Birth = 5 Nov 97  
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93
```

5. insert - insert 4th student data from the "StudentData.xlsx" file (Row5)

```
Roll_no = 101  
Name = Priyanka Chopra  
Date of Birth = 18 Jul 95  
Roll_no = 102  
Name = Virat Kohli  
Date of Birth = 5 Nov 97  
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93  
Roll_no = 104  
Name = Narendra Modi  
Date of Birth = 17 Sep 95
```

6. delete (roll number 102) - delete the student node with roll number 102

```
Roll_no = 101  
Name = Priyanka Chopra  
Date of Birth = 18 Jul 95  
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93  
Roll_no = 104  
Name = Narendra Modi  
Date of Birth = 17 Sep 95
```

7. delete (roll number 101) - delete the student node with roll number 101

```
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93  
Roll_no = 104  
Name = Narendra Modi  
Date of Birth = 17 Sep 95
```

8. insert - insert 5th student data from the "StudentData.xlsx" file (Row6)

```
Roll_no = 102  
Name = Arijit Singh  
Date of Birth = 25 Apr 94  
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93  
Roll_no = 104  
Name = Narendra Modi  
Date of Birth = 17 Sep 95
```

9. insert - insert 6th student data from the "StudentData.xlsx" file (Row7)

```
Roll_no = 101  
Name = Amit  
Date of Birth = 19 Sep 88  
Roll_no = 102  
Name = Arijit Singh  
Date of Birth = 25 Apr 94  
Roll_no = 103  
Name = Rakesh Kumar Bhadauria  
Date of Birth = 15 Jun 93  
Roll_no = 104  
Name = Narendra Modi  
Date of Birth = 17 Sep 95
```

10. insert - insert 7th student data from the "StudentData.xlsx" file (Row8)

11. print - print the linked list with the roll number, name and D.O.B

```
Roll_no = 101
Name = Amit
Date of Birth = 19 Sep 88
Roll_no = 102
Name = Arijit Singh
Date of Birth = 25 Apr 94
Roll_no = 103
Name = Rakesh Kumar Bhadauria
Date of Birth = 15 Jun 93
Roll_no = 104
Name = Narendra Modi
Date of Birth = 17 Sep 95
Roll_no = 105
Name = Amit
Date of Birth = 21 Apr 94
```

12. sort (name) - sort the name according to student names

```
Roll_no = 101
Name = Amit
Date of Birth = 19 Sep 88
Roll_no = 105
Name = Amit
Date of Birth = 21 Apr 94
Roll_no = 102
Name = Arijit Singh
Date of Birth = 25 Apr 94
Roll_no = 104
Name = Narendra Modi
Date of Birth = 17 Sep 95
Roll_no = 103
Name = Rakesh Kumar Bhadauria
Date of Birth = 15 Jun 93
```

13. print - print the linked list with the roll number, name and D.O.B

```
Roll_no = 101
Name = Amit
Date of Birth = 19 Sep 88
Roll_no = 105
Name = Amit
Date of Birth = 21 Apr 94
Roll_no = 102
Name = Arijit Singh
Date of Birth = 25 Apr 94
Roll_no = 104
Name = Narendra Modi
Date of Birth = 17 Sep 95
Roll_no = 103
Name = Rakesh Kumar Bhadauria
Date of Birth = 15 Jun 93
```


14. modify (roll number 103) - modify the student node having roll number 103

15. print - print the linked list with all the records for each student

```
104
Roll_no = 101
Name = Amit
Date of Birth = 19 Sep 88
Address = G-21, Sector 9, Opposite of Community House, Chandigarh
Phone no. = 9644258744
Roll_no = 105
Name = Amit
Date of Birth = 21 Apr 94
Address = 391, 3rd Floor, Aram Nagar Part 2, Andheri, Mumbai
Phone no. = 7874102563
Roll_no = 102
Name = Arijit Singh
Date of Birth = 25 Apr 94
Address = 181, 1st Floor, Aram Nagar Part 2, Andheri, Mumbai
Phone no. = 9874102563
Roll_no = 104
Name = Narendra Modi
Date of Birth = 17 Sep 95
Address = Parliament House, Room No.1, South Block, Raisina Hills, New Delhi
Phone no. = 9630258741
Roll_no = 103
Name = Rakesh Kumar Bhaduria
Date of Birth = 15 Jun 93
Address = Vice Chief of the Air Staff, Air Headquarters, New Delhi
Phone no. = 9876543210
```

Total Runtime:

real 0m0.003s

user 0m0.000s

sys 0m0.003s

Problem Statement 2:

Write a C Program for resizable deque using dynamic memory allocation, where a deque can perform the insertion and deletion operations at its both ends. The capacity of the deque depends on the number of elements currently stored in it, according to the following two rules:

- If an element is being inserted into a deque, when it is already full, then its capacity is doubled of its current size.

- After removing an element from a deque, if the number of elements are equal to half of the capacity of the deque, then its capacity is made half of its current size.

The program should have the following three functions: insert(), delete() and print().

The function print() should display the current size of the deque (capacity of deque) in terms of number of bytes.

Data Structures used:

1. Circular List for managing the entries

Dynamic memory allocation has been used.

Run time:

```
real 0m3.297s
user 0m0.000s
sys 0m0.002s
```

Snapshots of code running:
(Using the driver program)

```
psk@predator:~/Desktop/L1_18114038/src/Problem 2$ gcc prob2.c
psk@predator:~/Desktop/L1_18114038/src/Problem 2$ ./a.out
Enter the initial capacity of deque.
10
having roll number 103
15. print - print the linked list with all the records
Deque after rear insertion:
15
Note: In 'modify' function, the programmer can update
other fields except the roll number of
Deque after front insertion:
21 15
student.

Deque after rear insertion:
21 15 19
1. Queue Data structure: Used to hold the discarded ro
Deque after front deletion:
15 19
2. Circular Doubly Linked List: Used to hold the nodes
Deque after rear deletion:
15
```

Problem Statement 3:

Given three 2D arrays (for red, green and blue color pixels) of a digital image. For a particular image pixel, the color shade of that pixel is Red if the pixel value at that position of the matrix corresponding to RED is greater than that of GREEN and BLUE. Same goes for GREEN and BLUE shades also. Write a C program that can perform following operations on the given image file:

1. Remove all Red shades.
2. Remove all Green shades.
3. Remove all Blue shades.
4. RedOnly: Preserve any red shades in the image, but remove all green and blue.
5. GreenOnly: Preserve any green shades in the image, but remove all red and blue.
6. BlueOnly: Preserve any blue shades in the image, but remove all red and green.

Write a function pixelValue() that has x and y as two parameters and displays the current pixel (RED, GREEN and BLUE) values of the input image at the point with coordinates (x, y), where x and y are the row and column numbers in that image file, respectively.

Perform the testing of your code with the following TestCases:

Input: Q3_ip_Red.dat, Q3_ip_Green.dat and Q3_ip_Blue.dat are the three files with red, green and blue pixel values for the image file Q3_ip.jpg
Output after removing only green: Q3_op_Red.dat, Q3_op_Green.dat and Q3_op_Blue.dat are the output files w.r.t. red, green and blue pixel values after removing green from the input file Q3_ip.jpg, respectively.

Snapshots of code running:
(Using the driver program)

```
psk@predator:~/Desktop/L1_18114038/src/Problem 3$ gcc prob3.c
psk@predator:~/Desktop/L1_18114038/src/Problem 3$ ./a.out
Height of Image = 953
Width of Image = 1268
RGB values are as follows:
RED=254
GREEN=0
BLUE=0
```

The generated output files have also been attached.

Run time:

```
real 0m0.841s
user 0m0.813s
sys 0m0.029s
```