Name : Harshit Verma

Enrollment number : 18114029

Email : hverma1@cs.iitr.ac.in

Phone-number : +91 7905362294

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.

Test-case :

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

Problem Statement 2:

Write a C Program for resizeable 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

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:
● Remove all Red shades.
● Remove all Green shades.
● Remove all Blue shades.
● RedOnly: Preserve any red shades in the image, but remove all green and blue.
● GreenOnly: Preserve any green shades in the image, but remove all red and blue.
● 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

---

ALGORTIHMS AND DATA STRUCTURES USED :

1 >

    Algorithm : Bubble Sort and Sequencial Searching.
    Data Structure : Circular doubly linked list, Queue.

2 >

    Data Structure : Deque.

3 >
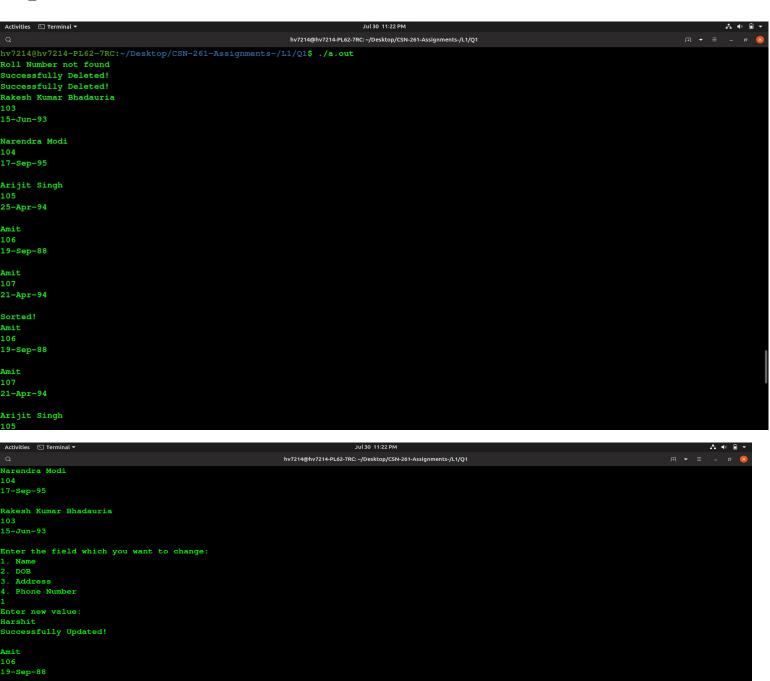
    Data Structure : Dynamic array.

SNAPSHOTS:

1 >

```
hv7214@hv7214-PL62-7RC:~/Desktop/CSN-261-Assignments-/L1/Q1$ ./a.out
Roll Number not found
Successfully Deleted!
Successfully Deleted!
Rakesh Kumar Bhadauria
103
15-Jun-93

Narendra Modi
104
17-Sep-95

Arijit Singh
105
25-Apr-94

Amit
106
19-Sep-88

Amit
107
21-Apr-94

Sorted!
Amit
106
19-Sep-88

Amit
107
21-Apr-94

Arijit Singh
105
```

```
Narendra Modi
104
17-Sep-95

Rakesh Kumar Bhadauria
103
15-Jun-93

Enter the field which you want to change:
1. Name
2. DOB
3. Address
4. Phone Number
1
Enter new value:
Harshit
Successfully Updated!

Amit
106
19-Sep-88

Amit
107
21-Apr-94

Arijit Singh
105
25-Apr-94

Narendra Modi
104
17-Sep-95

Harshit
```

```
15-Jun-93

Enter the field which you want to change:
1. Name
2. DOB
3. Address
4. Phone Number
1
Enter new value:
Harshit
Successfully Updated!

Amit
106
19-Sep-88

Amit
107
21-Apr-94

Arijit Singh
105
25-Apr-94

Narendra Modi
104
17-Sep-95

Harshit
103
15-Jun-93

CPU Time: 0.000461
hv7214@hv7214-PL62-7RC:~/Desktop/CSN-261-Assignments-/L1/Q1$ █
```

2 >

```
hv7214@hv7214-PL62-7RC:~/Desktop/CSN-261-Assignments-/L1/Q2$ ./a.out
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
1
Enter number:
14
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
2
Enter number:
13
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
1
Enter number:
1
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
```

```
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
2
Enter number:
12
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
4
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
5
13 14 1
-----------------MAIN-MENU-----------------
1. Push back
2. Push front
3. Pop back
4. Pop front
5. Print
6. Exit
6
CPU Time: 0.000104hv7214@hv7214-PL62-7RC:~/Desktop/CSN-261-Assignments-/L1/Q2$
```

3 >


```
-----------MAIN MENU-----------
1. Remove red shades
2. Remove green shades
3. Remove blue shades
4. Preserve red
5. Preserve green
6. Preserve blue
7. Pixel Value
8. Print color component values
9. Exit
2
Done successfully.
-----------MAIN MENU-----------
1. Remove red shades
2. Remove green shades
3. Remove blue shades
4. Preserve red
5. Preserve green
6. Preserve blue
7. Pixel Value
8. Print color component values
9. Exit
7
952 1267
255 255 255
Done successfully.
-----------MAIN MENU-----------
1. Remove red shades
2. Remove green shades
3. Remove blue shades
4. Preserve red
5. Preserve green
6. Preserve blue
7. Pixel Value
8. Print color component values
```