

CSN 261

DATA STRUCTURES LAB

**ASSIGNMENT 1**

DIVYANSHU SETIA

18114020

B. TECH, CSE

## PROBLEM 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.

## INPUT:

```
delete(108);
printf("\n");
insert(Data[0].name, Data[0].dob, Data[0].address, Data[0].phoneNo);
insert(Data[1].name, Data[1].dob, Data[1].address, Data[1].phoneNo);
insert(Data[2].name, Data[2].dob, Data[2].address, Data[2].phoneNo);
insert(Data[3].name, Data[3].dob, Data[3].address, Data[3].phoneNo);
delete(102);
delete(101);
insert(Data[4].name, Data[4].dob, Data[4].address, Data[4].phoneNo);
insert(Data[5].name, Data[5].dob, Data[5].address, Data[5].phoneNo);
insert(Data[6].name, Data[6].dob, Data[6].address, Data[6].phoneNo);
print();
printf("\n");
sort();
print();
printf("\n");
struct student* modify103 = (struct student*) malloc(sizeof(struct student));
strcpy(modify103->name, "Taylor Swift");
strcpy(modify103->dob, "12-Jan-45");
strcpy(modify103->address, "Brooklyn 99");
modify103->phoneNo = 1489302673;
modify(103, *modify103);
print();
```

## OUTPUT:

```

List is empty

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

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

```

```

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: Taylor Swift
Date of Birth: 12-Jan-45

```

## ALGORITHMS AND DATA STRUCTURES:

- A circular doubly linked list is implemented for storing student data along with a dequeue for storing deleted roll numbers.

- The csv file is read using fgets()
- Sorting is done with the help of bubble sort and strcmp()

## TIME:

```
real    0m0.212s
user    0m0.000s
sys     0m0.016s
```

---

## PROBLEM 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.

## INPUT:

```
insertLeft(1);
print();
insertLeft(2);
print();
insertRight(3);
print();
insertLeft(4);
print();
insertRight(5);
print();
deleteRight();
print();
deleteLeft();
print();
deleteLeft();
print();
```

## OUTPUT:

```
1
Size is: 4
1
2
Size is: 8
1
2
0
3
Size is: 16
1
2
4
3
```

```
Size is: 16
1
2
4
0
0
0
5
3
Size is: 32
1
2
4
3
Size is: 16
1
2
0
3
Size is: 16
1
3
Size is: 8
```

## ALGORITHMS AND DATA STRUCTURES:

- Arrays are used to implement doubly ended queue.
- `calloc()` function is used for dynamic memory allocation

## TIME:

```
real    0m0.169s
user    0m0.000s
sys     0m0.031s
```

---

## PROBLEM 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.

## INPUT:

```
pixelValue(145,67);
pixelValue(1245,497);
pixelValue(907,43);

removeAllShades(red);

pixelValue(145,67);
pixelValue(1245,497);
pixelValue(907,43);
return 0;
```

## OUTPUT:

```
Red: 254
Blue: 0
Green: 0

Red: 0
Blue: 0
Green: 1

Red: 46
Blue: 181
Green: 117

Red: 0
Blue: 0
Green: 0

Red: 0
Blue: 0
Green: 1

Red: 46
Blue: 181
Green: 117
```

## ALGORITHMS AND DATA STRUCTURES:

- A static 3-D global array is defined to store data.
- fscanf() is used to read the files
- Nested for loops are used in functions

## TIME:

```
real    0m0.646s
user    0m0.516s
sys     0m0.078s
dootia@DESKTOP-DTT
```