Indian Institute of Technology Roorkee
Department of Computer Science and Engineering

CSN-261: Data Structures Laboratory (Autumn 2019-2020)

Lab Assignment – 1

Name: Hemil Sanjaybhai Panchiwala

Enrollment No.: 18114031

Branch: Computer Science & Engineering

Sub Batch: O2

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.

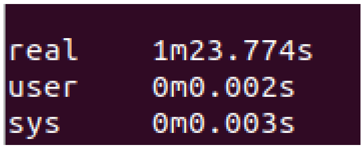
Data Structures used :-

Doubly Circular Linked List, Queue, Array

Algorithm

- Bubble Sort is used to sort the doubly circular linked list.
- Linear Search is used to search particular node in doubly circular linked list.
- Here, we have extracted the data from csv file and then created a doubly circular linked list and have performed different operations in doubly circular linked list: - Insert
    - Delete
    - Modify
    - Sort
    - Search
    - Print
- Then we insert the deleted roll no. in the queue for its use in further insertion.

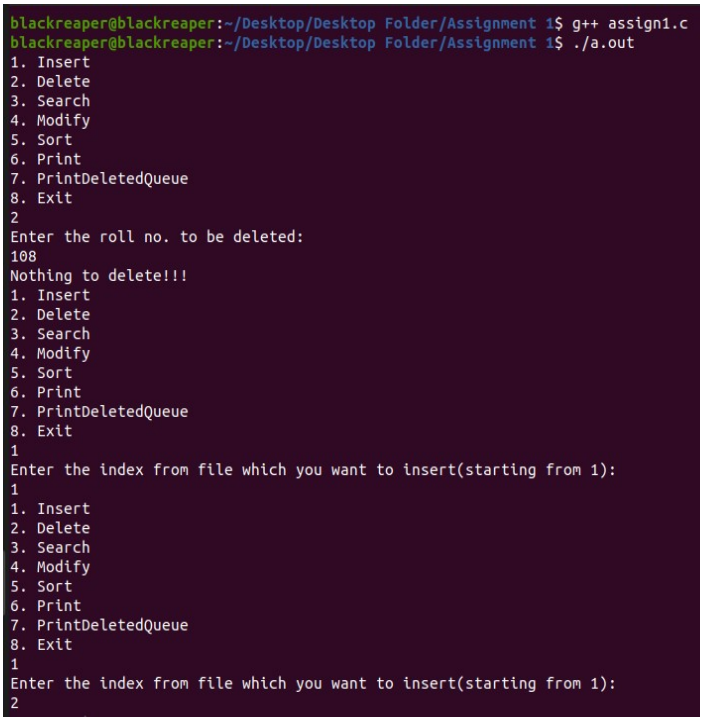
Time taken:



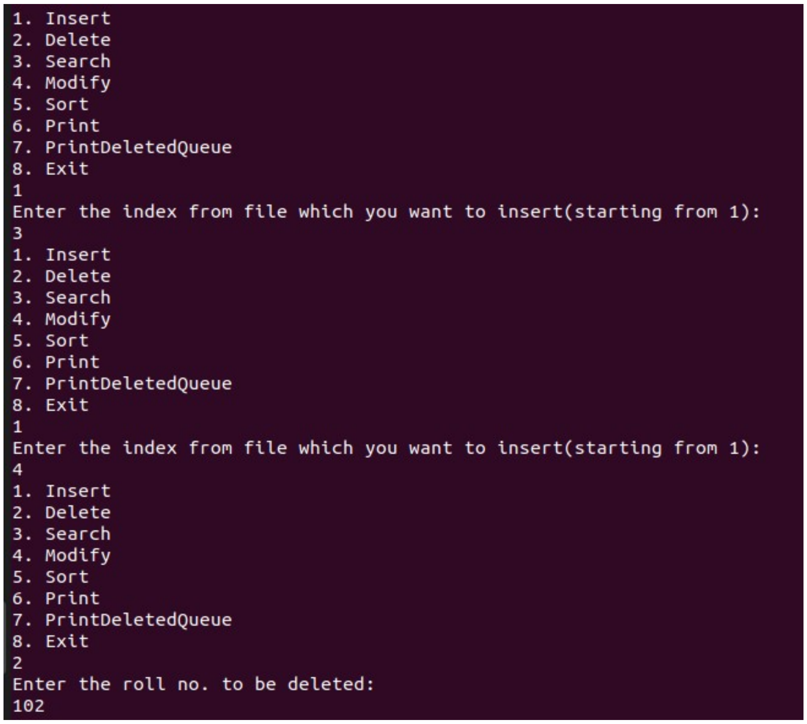
```
real	1m23.774s
user	0m0.002s
sys	0m0.003s
```

Note: real shows the complete time taken including user time in the program.

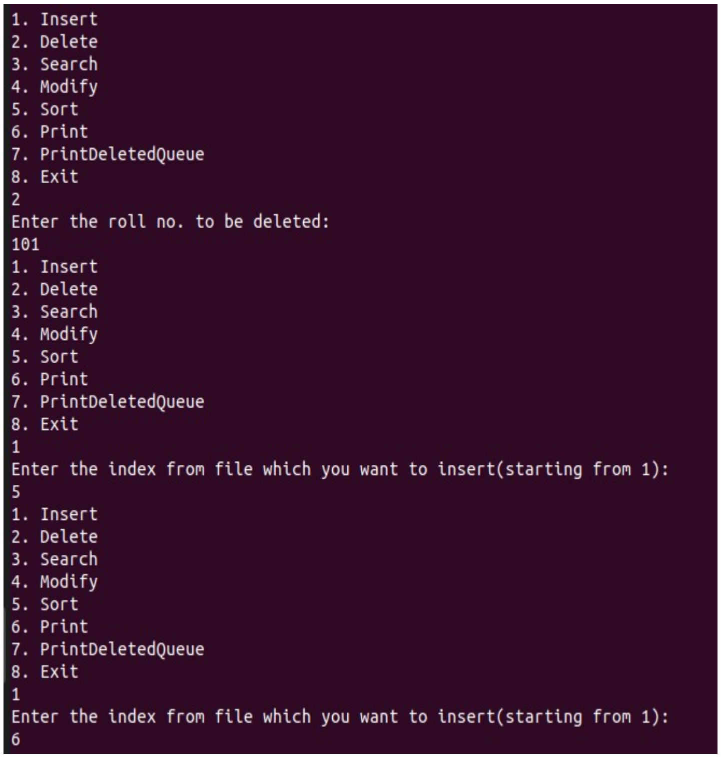
Snapshots



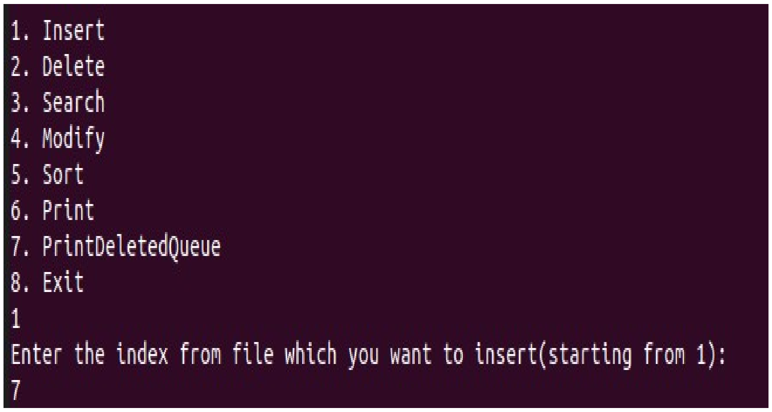
```
blackreaper@blackreaper:~/Desktop/Desktop Folder/Assignment 1$ g++ assign1.c
blackreaper@blackreaper:~/Desktop/Desktop Folder/Assignment 1$ ./a.out
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
2
Enter the roll no. to be deleted:
108
Nothing to delete!!!
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
1
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
2
```



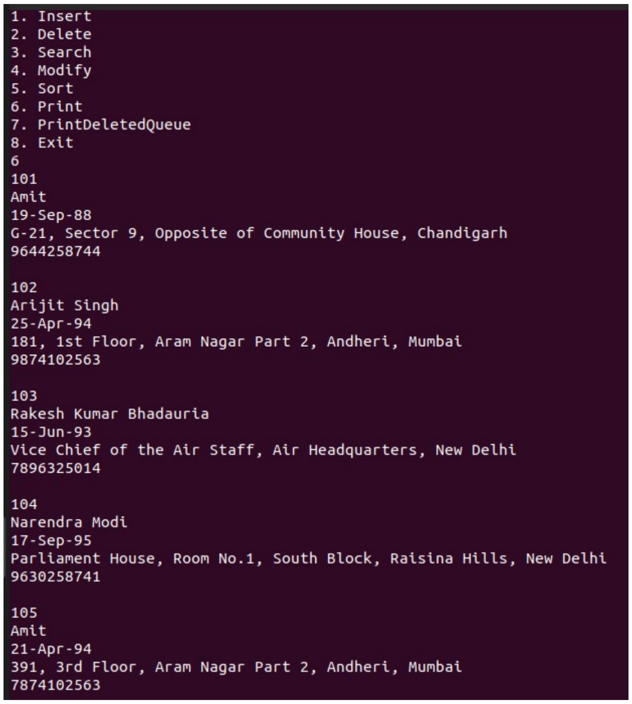
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
3
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
4
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
2
Enter the roll no. to be deleted:
102
```



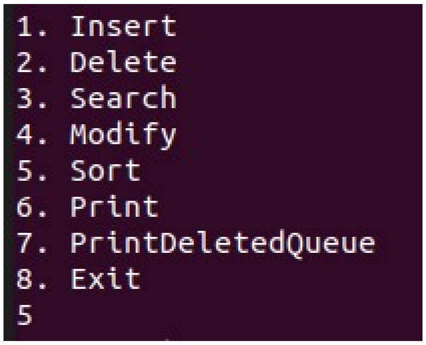
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
2
Enter the roll no. to be deleted:
101
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
5
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
6
```



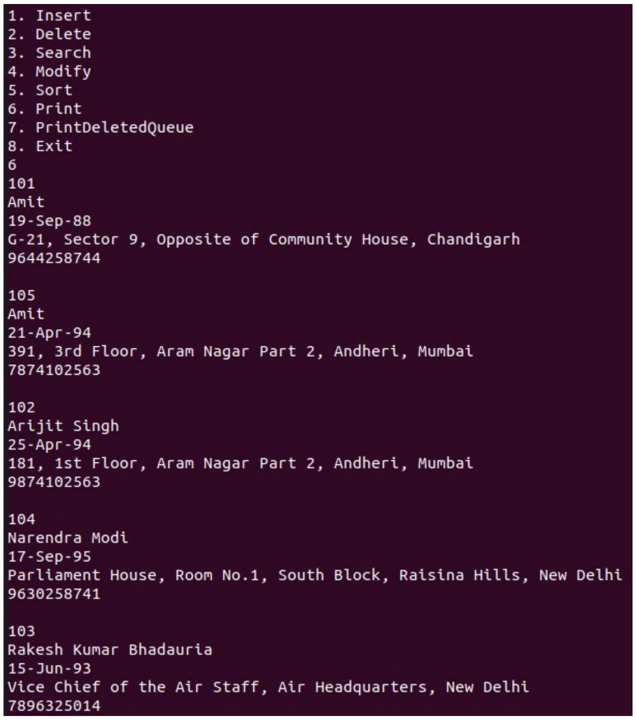
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
1
Enter the index from file which you want to insert(starting from 1):
7
```



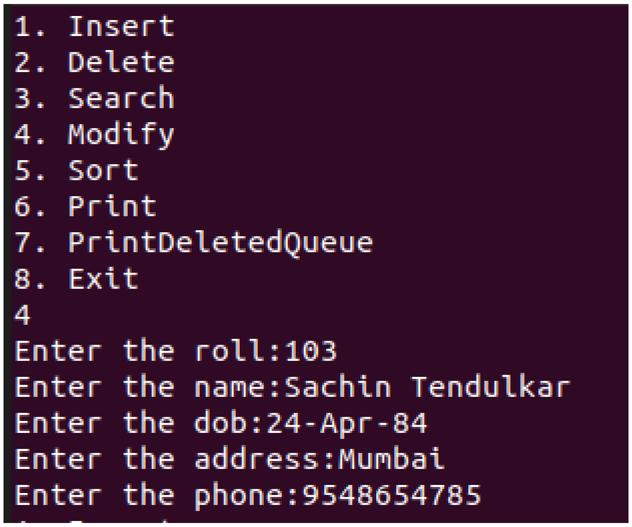
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
6
101
Amit
19-Sep-88
G-21, Sector 9, Opposite of Community House, Chandigarh
9644258744

102
Arijit Singh
25-Apr-94
181, 1st Floor, Aram Nagar Part 2, Andheri, Mumbai
9874102563

103
Rakesh Kumar Bhadauria
15-Jun-93
Vice Chief of the Air Staff, Air Headquarters, New Delhi
7896325014

104
Narendra Modi
17-Sep-95
Parliament House, Room No.1, South Block, Raisina Hills, New Delhi
9630258741

105
Amit
21-Apr-94
391, 3rd Floor, Aram Nagar Part 2, Andheri, Mumbai
7874102563
```



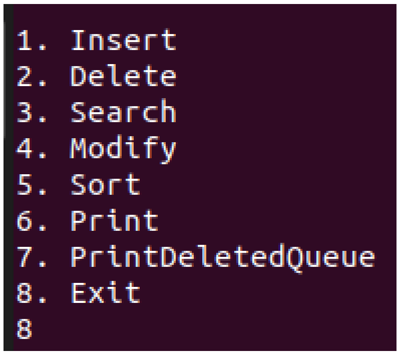
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
5
```



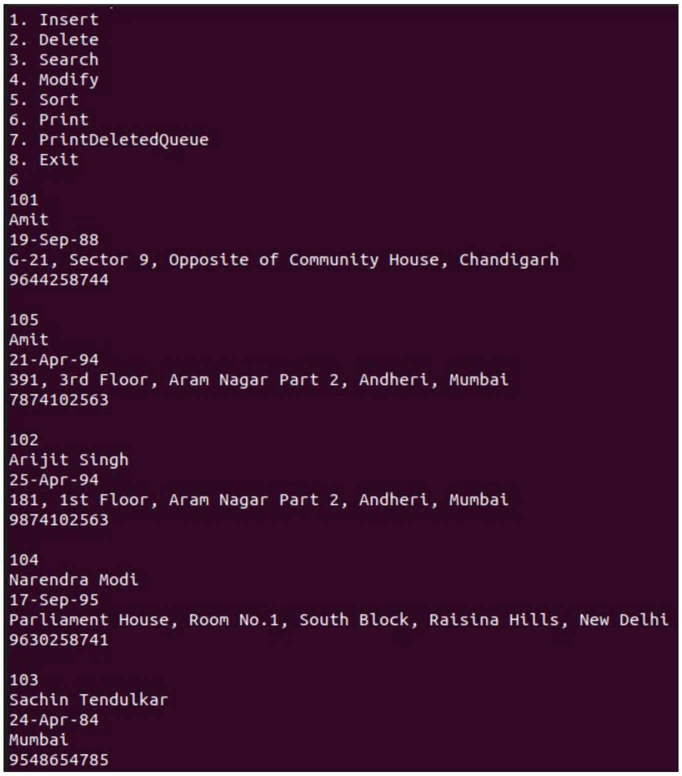
```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
6
101
Amit
19-Sep-88
G-21, Sector 9, Opposite of Community House, Chandigarh
9644258744

105
Amit
21-Apr-94
391, 3rd Floor, Aram Nagar Part 2, Andheri, Mumbai
7874102563

102
Arijit Singh
25-Apr-94
181, 1st Floor, Aram Nagar Part 2, Andheri, Mumbai
9874102563

104
Narendra Modi
17-Sep-95
Parliament House, Room No.1, South Block, Raisina Hills, New Delhi
9630258741

103
Rakesh Kumar Bhadauria
15-Jun-93
Vice Chief of the Air Staff, Air Headquarters, New Delhi
7896325014
```



```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
4
Enter the roll:103
Enter the name:Sachin Tendulkar
Enter the dob:24-Apr-84
Enter the address:Mumbai
Enter the phone:9548654785
```



```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
8
```



```
1. Insert
2. Delete
3. Search
4. Modify
5. Sort
6. Print
7. PrintDeletedQueue
8. Exit
6
101
Amit
19-Sep-88
G-21, Sector 9, Opposite of Community House, Chandigarh
9644258744

105
Amit
21-Apr-94
391, 3rd Floor, Aram Nagar Part 2, Andheri, Mumbai
7874102563

102
Arijit Singh
25-Apr-94
181, 1st Floor, Aram Nagar Part 2, Andheri, Mumbai
9874102563

104
Narendra Modi
17-Sep-95
Parliament House, Room No.1, South Block, Raisina Hills, New Delhi
9630258741

103
Sachin Tendulkar
24-Apr-84
Mumbai
9548654785
```

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.

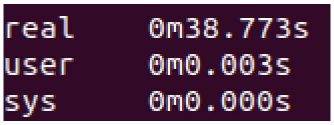
Data Structures used :-

Dequeue, Array

Algorithm

- **Initially, a deque is created dynamically with particular size and then we are performing different operations like insertion and deletion from left and right side.**
- **Now if the deque is filled then using realloc function the size of deque is doubled and if no. of elements in deque is less than half then again through realloc function the size of deque is halved.**
- **The deque here is made through array.**

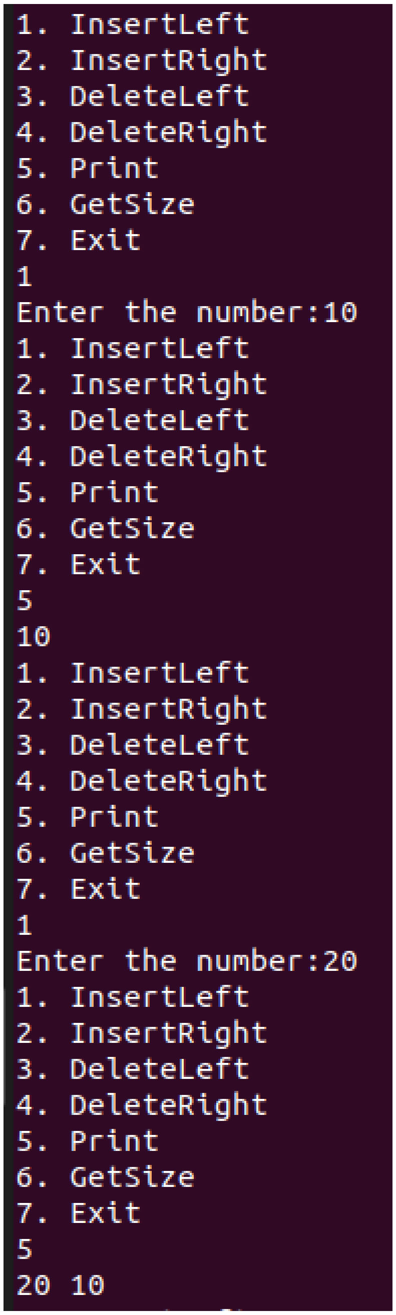
Time taken:



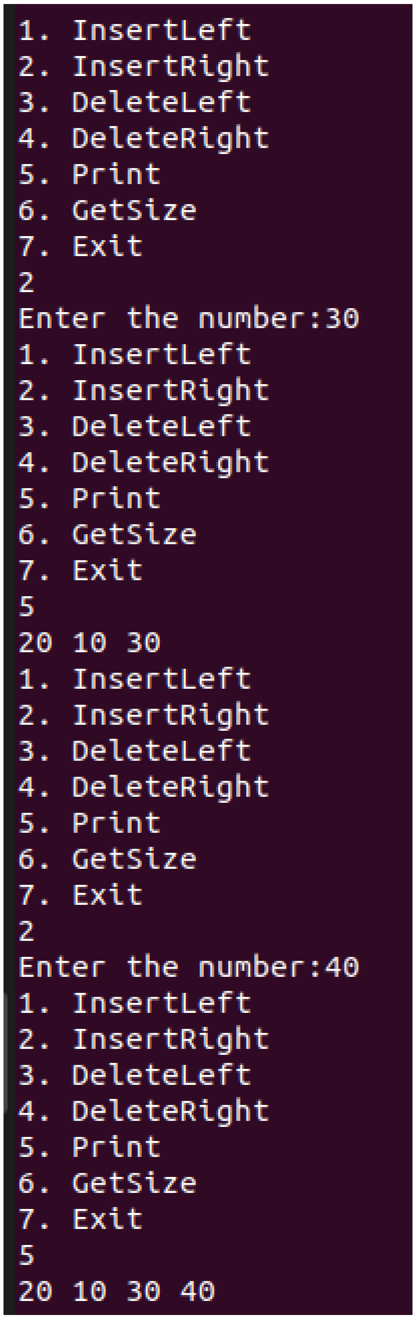
```
real	0m38.773s
user	0m0.003s
sys	0m0.000s
```

Note: real shows the complete time taken including user time in the program.

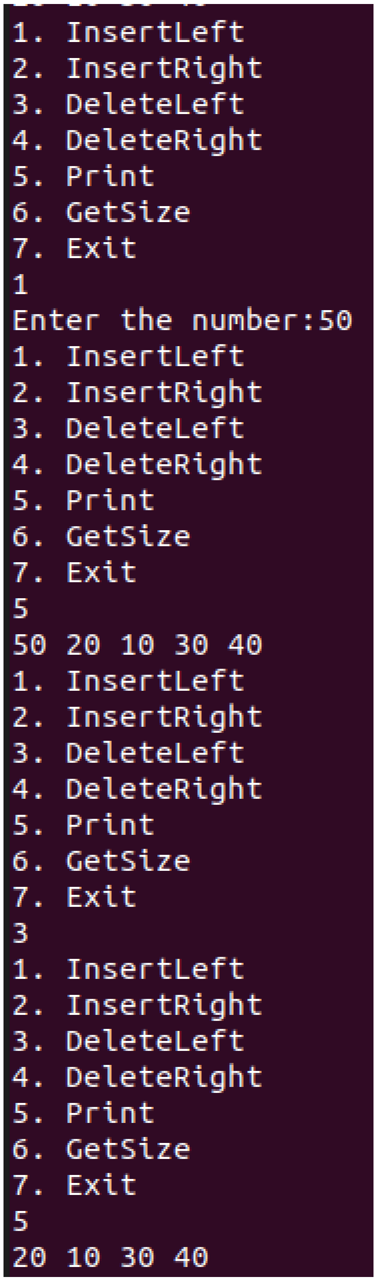
Snapshots



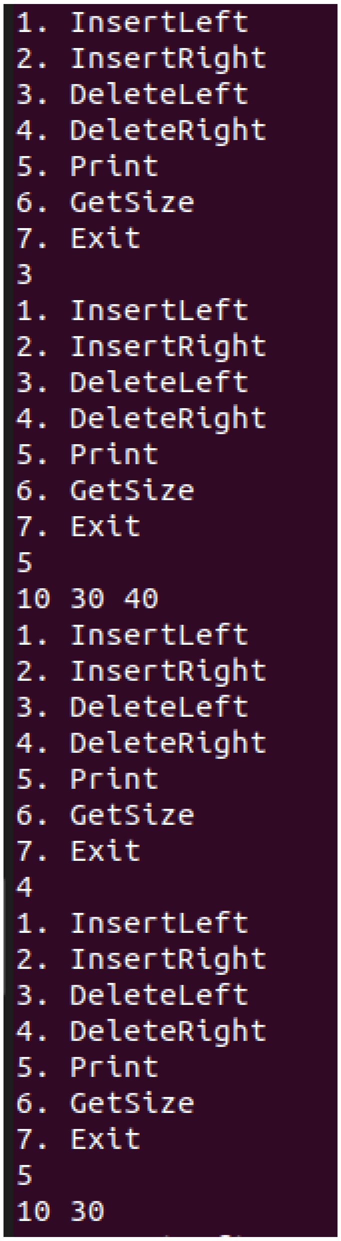
```
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
1
Enter the number:10
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
10
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
1
Enter the number:20
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
20 10
```



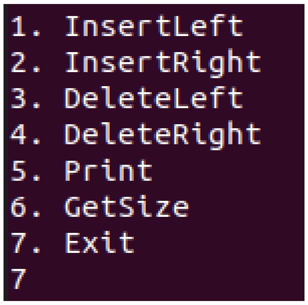
```
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
2
Enter the number:30
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
20 10 30
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
2
Enter the number:40
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
20 10 30 40
```



```
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
1
Enter the number:50
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
50 20 10 30 40
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
3
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
20 10 30 40
```



```
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
3
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
10 30 40
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
4
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
5
10 30
```



```
1. InsertLeft
2. InsertRight
3. DeleteLeft
4. DeleteRight
5. Print
6. GetSize
7. Exit
7
```

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.

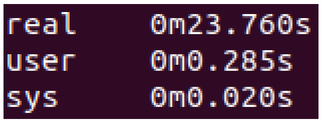
Data Structures used :-

Array

Algorithm

- **In this program, we first retrieve the total no. of pixels from the input image using the stb library.**
- **Then the arrays of intensity of colours are retrieved from the given input files of the colours.**
- **Then we have perform different functions given in question and then finally have printed the intensity of all three colours at a particular pixel.**

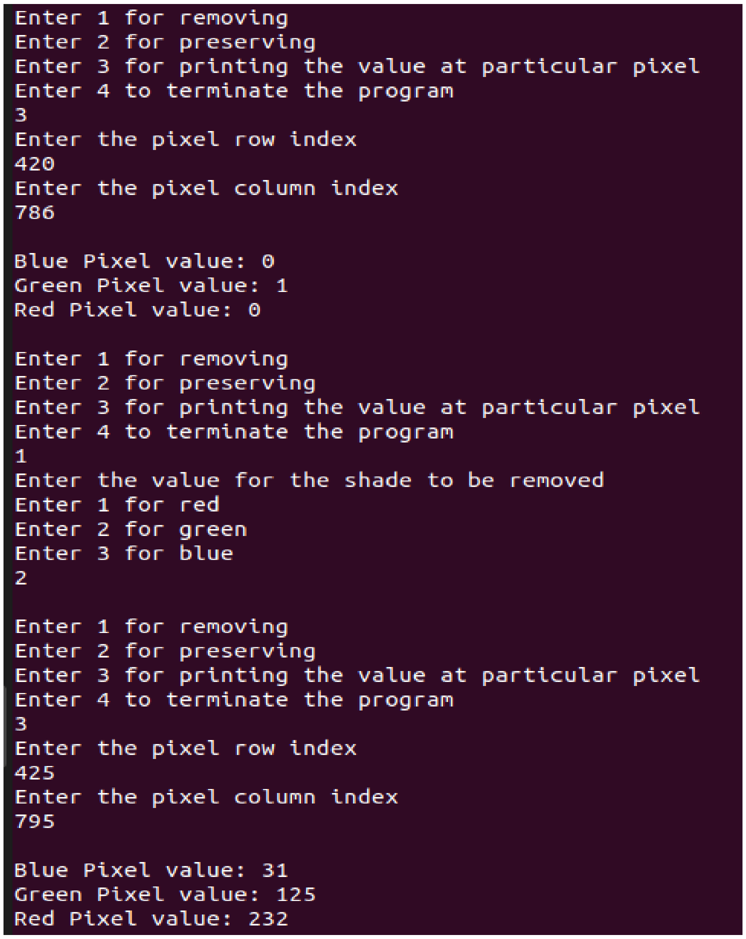
Time taken:



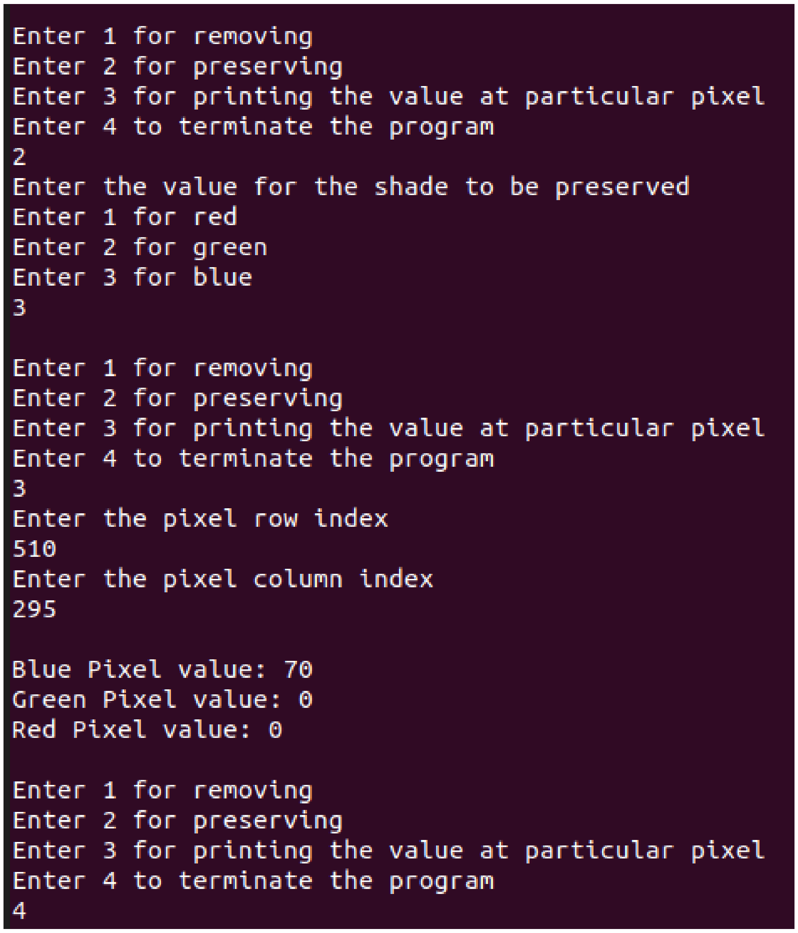
```
real	0m23.760s
user	0m0.285s
sys	0m0.020s
```

Note: real shows the complete time taken including user time in the program.

Snapshots



```
Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
3
Enter the pixel row index
420
Enter the pixel column index
786

Blue Pixel value: 0
Green Pixel value: 1
Red Pixel value: 0

Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
1
Enter the value for the shade to be removed
Enter 1 for red
Enter 2 for green
Enter 3 for blue
2

Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
3
Enter the pixel row index
425
Enter the pixel column index
795

Blue Pixel value: 31
Green Pixel value: 125
Red Pixel value: 232
```



```
Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
2
Enter the value for the shade to be preserved
Enter 1 for red
Enter 2 for green
Enter 3 for blue
3

Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
3
Enter the pixel row index
510
Enter the pixel column index
295

Blue Pixel value: 70
Green Pixel value: 0
Red Pixel value: 0

Enter 1 for removing
Enter 2 for preserving
Enter 3 for printing the value at particular pixel
Enter 4 to terminate the program
4
```