ASSIGNMENT 1

Data Structures Laboratory



**Rishi Chordia**

18118052

BTech CSE

# 

# 

# 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.

# Algorithms and Implementation:

# With the use of priority queues, we can keep a track on the roll no of new students

# We use a simple bubblesort algorithm with O(n\*n) complexity for sorting the linked list

# Other functions like insertion modificatoin deletion etc will take O(n) time

# File handling has been achieved through the fgets function.

# A Standard data structure,doubly linked list, has been used to store the student details.

# GDB was used for debugging.

# 

# 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.

# Algorithms and Implementation:

# Deques are implemented using circular array as it is better than linked lists due to lower space and time complexity

# Deques are allocated memory dynamically using malloc function

# Insertion and deletion operations are carefully implemented keeping in mind all the special cases

# 

# 

# 

# 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 pixelvalue 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 displaysthe current pixel (RED, GREEN and BLUE) values of the input image at the point with coordinates (x,y), where xand y are the row and column numbers in that image file, respectively.

# Algorithms and Implementation

# Reading from the file is done using the scanf with , as delimiter and keeping a new line in mind

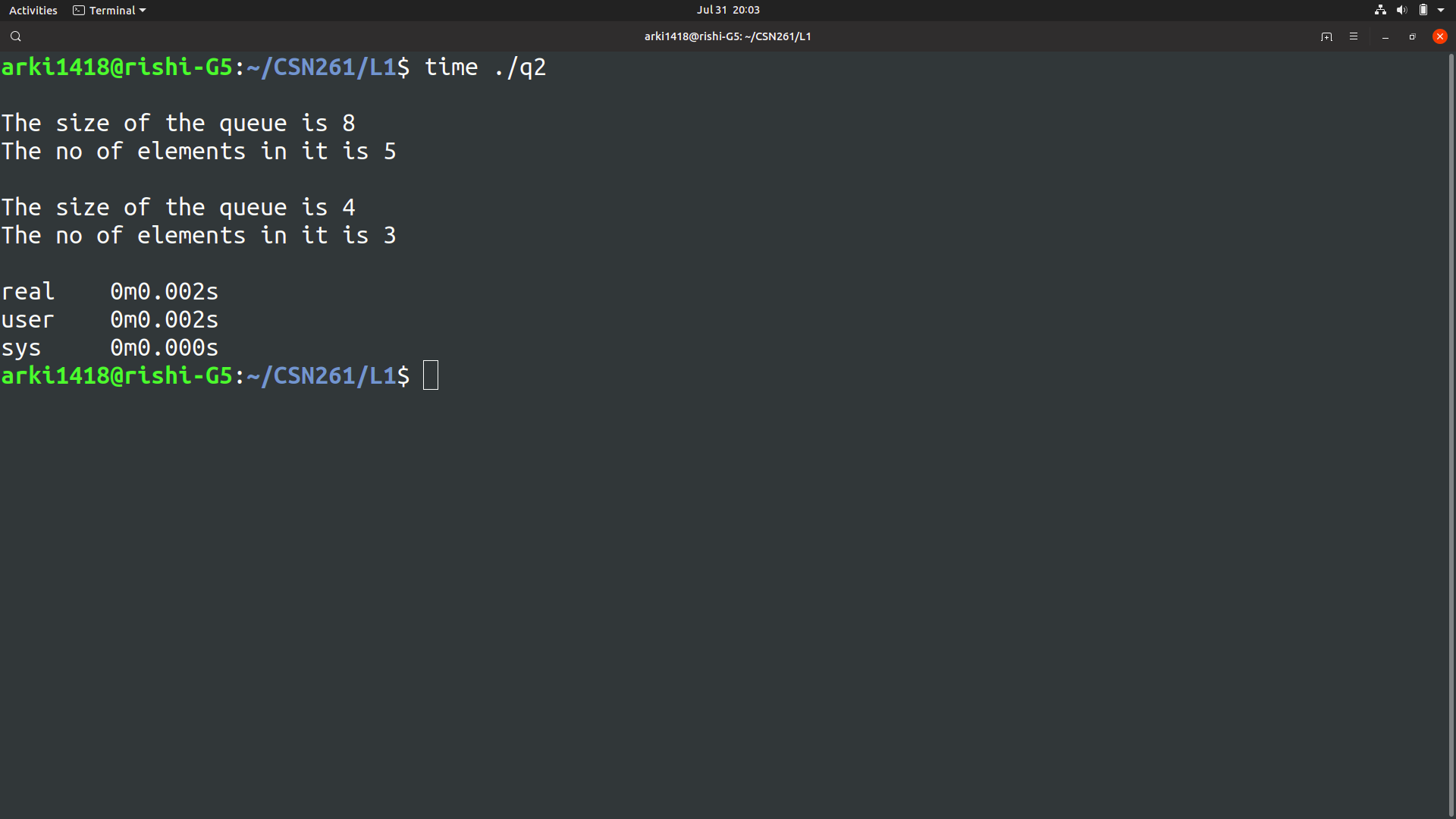
# Dynamic 2d arrays are used to store the data and cope up with the memory demand

# A user friendly interface is provided so that the user can perform any operation.

# GDB was used for debugging. Snapshots and Computation Time

# Question 1

Question 2:



Question 3:

