CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

In India traffic is the main huge problem for everyone, in order to control the traffic in India we use manually operated traffic light or signals that is how we can know how much time will it take to the vehicle when it need to stop at the signal based on the given lights .Whether the vehicle must stop or get ready or to move in the particular signal.

The most modern traffic signals were invented in America. New York had a three colour system in 1918 that was manually from a tower by the middle of the street. In 1923 he invented an electric traffic light system for the better convince using a pole with a cross section on which the words STOP and GO...

These are also named as stop lights roads traffic lamps, traffic signals, stop and go lights which are the main devices placed at road crossing for the people. Everyday pedestrian crossing and other locations to control competing flows to traffic in order to make it easy. This traffic system is designed to manage traffic signals timings based on the traffic on the corresponding road. Traffic lights have been fixed in the world. Traffic signs are basically categories into three types: regulatory, warning, and guide signs.

The shape and colour of signs indicate the type of information a sign will contain and what to do when the colour shows.

The commonly used traffic signs:

- A red STOP sign means you must make a full stop before entering the driving path or crossing the road.
- A yellow WAIT signs mean you must wait till the signals move from the colour.
- A green GO signs means you must move when the signal appears.

When the signals start reducing collisions and waiting time for both vehicles and pedestrians it avoids of making collision. Encouraging travel within the speed limit to meet green lights is very risky for the people. Reducing unnecessary stopping and starting of traffic - this in turn reduces fuel consumption, accidents, air and noise pollution, and wear and tear of the vehicles.

- Reducing travel time and increasing of the patients.
- Reducing driver frustration and road range accidents from the people safety.

Traffic signal control is very important and the most challenging real world problem which mainly aims to minimize the travel time of the vehicles by coordinating their movements at the road for the main intersection points. These negative outcomes lead to only grow of the overtime as more people are been there.

1.2 OBJECTIVES:

To achieve safe measures and movement by reducing accident levels. It promotes and accommodates the maintenance and improvements of public transport. To restrain traffic speeds and safeguard the environment we have to keep the signals in order to make it not happen.

The project's overall objective is to develop a traffic signal controller, which can be sensitive to changing traffic management policy, and aim to achieve a wide range of transport and environmental objectives in addition to minimization of vehicle delays and speeds control by the signal.

In this we can develop a multiagent traffic light system based on the multi objective by making the decision as a frame work. In order in the response of the effectively of the changing the environment and non environment or stationary of the network in the traffic light the open source of the Green light district vehicles as the most test bed frameworks .

These main traffic signals are one of the main traffic management's tools that are mainly used to the control of the traffic.

1.3 METHODOLOGY

In this project I am using data structures as a main topic to determine how to use the traffic signal control.
The main object of my project is how I can control the traffic and accidents in order to save the people life. And how to access the traffic signals for exiting the preemption control.
It provides orderly movement of intersection of the signals which have the ability to be flexible for changes in the traffic flow.
In this project I will check the normal function of the traffic signals which requires more slight control and coordination to ensure. Seeing this people shows the main aim of the traffic signals.

1.4 EXPECTED OUT COMES

The outcomes of my project include the main objective of traffic signal system this involves in the data structure which mainly involves in circular queue.

Red –to stop

Yellow-to gets ready

Green-to go

1.4.1. Advantages:

Traffic control signals provide for an orderly movement of traffic.

They help in reducing the popularity of an accident of some special nature.

They intercept heavy traffic to allow other the people to cross the road in safety way.

They provide authority to the drivers to move with confidence in the roads.

They control the speed of vehicles on main roads as well as on secondary roads.

1.4.2. Disadvantages:

Traffic control signals may result in a collision of vehicles and accidents.

They may cause a slow delay in the quick movement of traffic.

1.5.1. HARDWARE REQUIRMENTS

- Processor : Any Processor above 500 MHz
- RAM : 512
- Hard Disk : 10 GB
- Input device : Standard Keyboard and Mouse
- Output device : VGA and High Resolution Monitor
- Software : turbo C
- Compiler :dev c++
- Operating system :windows

1.5.2. SOFTWARE REQUIREMENTS

- Operating system : Windows XP
- Front End : ASP.Net 2.0
- Server :Internet Information Services

CHAPTER – 2

DATA STRUCTURES

Data structure is used for storing data in a format.

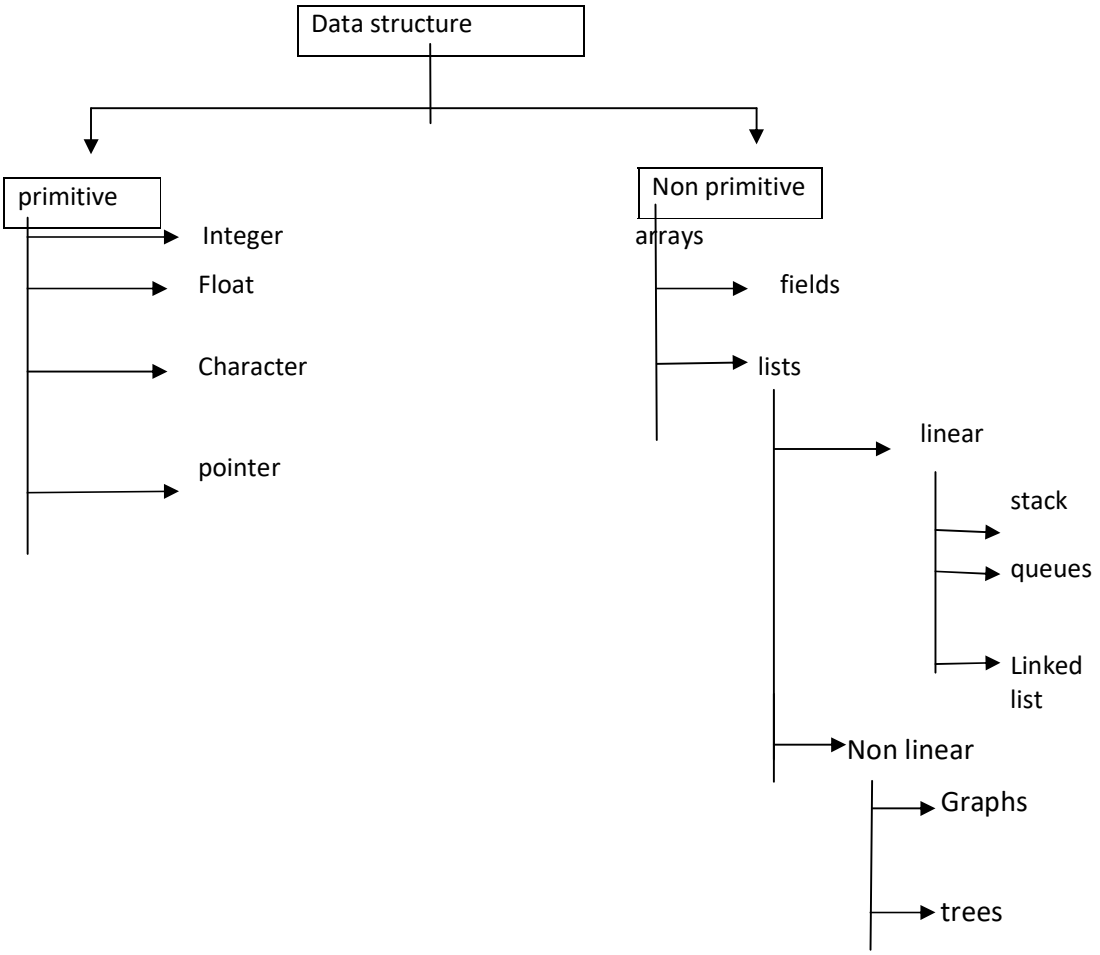

fig 2.1 Representation of Data structure

Data structure:

It is a data of organization and managements and storage format that enables efficient accesses and for the modification .More precisely a data structure is a collection of the data values and elements in the data the relationship among them and the functions or preparations that can be applied in the data.

Data structures are of two types: primitive data structures and non-primitive data structures.

Primitive data structures can be directly manipulated by machine instructions.

Primitive data structure are those which are predefined way of storing the values in the system .This primitive data structure are char , int, float ,double or the pointer .Character are mainly considering all the internally stored values as the way int and float both falls under double and they are predefined as addition and subtraction of the given elements

Examples: 1.int

2. Float
3. Char
4. Pointer

Int: int is the other name for integer in which all the data elements are stored in integer values.

Integral values may be different from the sizes and the y may or may not allow the negative values which are commonly represented in the data.

Float: Float is the term which is mainly used in the various programming languages to define a particular variable we use float it is used to create a numbers using the float variables in the declaration of having digits on the both sides at the decimal points. This is based on the integer data type.

Char: Char it is a short form of the character in which the data type that holds one character (letters, numbers, etc) of the elements. For example the value of the char can be any one-character value such as “A”,”4” or #

Pointer: Pointer is used to address the values stored anywhere in the memory given for the data. To obtain the values stored at the location is also known as dereferencing of the pointer. Pointer improves the performance for the elements stored in the data.

Non-primitive data structures cannot be directly manipulated using machine instructions.

Non primitive data structure is a data type in which something else is known such as an array structure or the class is mainly known as the non primitive data structure. The data type that are mostly derived from the primary data type this are also known as non primitive data structure of the elements This non primitive data types are mainly used to store the group of data.

Examples: 1.Arrays

2. Lists

3. Files

Arrays: An arrays is the data structure or the simply an arrays is the data in the structure consisting of the collection of data, elements. At least one array index or the key. An array is stored in the data can be computed from the index.

Types of arrays:

- Single dimension array
- Multi-dimension array

Single dimension array

SYNTAX:

data type array name [array size];

Where

data type=type of the data (int, char, float)

array name=name of the array

Array size=size of the array

EXAMPLE:

Int a [30];

Where

Int is the data type

A is the variable name

30 is the size of the array

Multi-dimension array:

SYNTAX:

data type array name [row size] [column size];

Where

data type= type of the data (int, char, float)

array name=name of the array

row size=size of the row

column size=size of the column

EXAMPLE:

Int a [30][20];

Where,

Int is the data type

a is the variable name

30 is the size of the row

20 is the size of the column

Lists: A list is a data type that mainly represents a countable number of ordered values where the same value may occur more than once at the data.

Files: A file is nothing but a collection of the records. This file usually implies that all the records are stored in the secondary storage of the internal storage of the data.

Lists are again classified into 1.Linear lists

2. Non-linear lists.

Linear lists:

The elements are mainly ordered in the linear list in the way of linear sequence. Linear lists are usually simply denoted as the part of the list. An array is a list of the data structure allowing all the insertion and deletion of the elements. So they are also called as dynamic data structures.

1. Linear list consists of: (a) Stack

(b) Queues

(c) Linked list.

Stack: A stack is a data type which shows all the collections of the elements with the main two principles of push means which add an element and pop which removes the elements which are added recently.

Different operations of stack are:

1. Push- to insert the elements into stack
2. Pop-to delete elements from the stack
3. Display-to display the elements present in the stack

EXAMPLE:

Just imagine that you have a cookies jar. in this jar, you can add one cookie. The process when you add one cookie inside the jar from top most position is called as push operation. Similarly, when you remove one cookie from the jar then that process of cookie removal from the top most position is called pop operation.

The last entered cookie into the cookie jar is the first to be removed from the jar, thus a stack flows a last in first out principle (LIFO).

Stack applications:

1. Recursion
    1. Factorial
    2. Fibonacci
    3. Towers of Hanoi
2. Conversion of expressions.
3. Evaluation of postfix expressions.

Queues: It is the collection of the data which the data of the intersection and deletion both happens at one end itself.

Different operations of queues are:

1. Insertion- to insert the elements into the queue through rare end.
2. Deletion-to delete the elements from the queue through front end.
3. Display-to display the elements present in the queue

Linked list: It is a data structure which is connected in the link Linked list is a sequence in which all the data elements are stored in the list which is also called as element. Each link which is connected in the linked list contains a link called as the next.

Size of the linked list is not fixed and data items can be added at any locations in the linked list. The disadvantage of the linked list is that if we require a particular node, then we must traverse from the first node to the particular node that we require.

In computer science, a linked list is a linear collection of data elements, whose order is not given by their physical placement in memory. Instead, each element points to the next. It is a data structure consisting of a collection of nodes which together represent a sequence. In its most basic form, each node contains: data and a reference (in other words, a link) to the next node in the sequence. This structure allows for efficient insertion or removal of elements from any position in the sequence during iteration. More complex variants add additional links, allowing more efficient insertion or removal of nodes at arbitrary positions. A drawback of linked lists is that access time is linear (and difficult to pipe line). Faster access is not feasible. Arrays has best cache locality compared to linked lists.

How to add elements to linked lists

You can add elements to beginning, middle or end of linked list.

Add to beginning

- Allocate memory for new node.
- Store data.
- Change next of new node to print the head.
- Change head to point to recently created node.

Add to end

- Allocate memory to the new node
- Store data.

Add to middle

- Allocate memory for new node.
- Store data for new node.
- Traverse to node just before the required position of new node.
- Change next pointer to include new node in between.

How to delete from a linked list

You can delete either from beginning, end or from a particular position.

Delete from beginning

- Point head to the second node.

Delete from end

- Traverse to second last element.
- Change its next pointer to NULL.

Delete from middle

- Traverse to element before the element to the deleted.
- Change next pointer to exclude the node from the chain

Basic operations of linked list

1. Insertion-to insert the data to the node of linked list.
2. Deletion-to delete the data of the node from the linked list.
3. Display-to display the data present in the node of linked list.

Types of linked list:

1. Single linked list
2. Double linked list
3. Circular linked list
4. Header linked list

1) SINGLE LINKED LIST

Single linked list is the linked list which has only one link. In other words, it is a collection of nodes where each node as two parts i.e., data which contain the elements and link to the next node.

It can traverse only in one direction. Single linked list consumes less memory. But the disadvantage of single linked list is that once the pointer is moved forward, it can't go back.

STRUCTURE REPRESENTATION OF SINGLE LINKED LIST:

```
struct slist
{
   int data;
  struct slist *ptr;
};
```

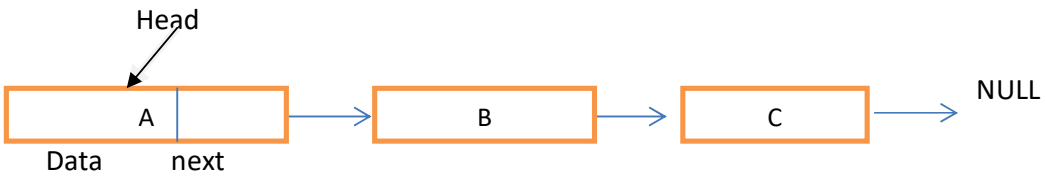

Fig2.2 Representation of single linked list

2) DOUBLE LINKED LIST:

Double linked list is a linked list which has two links. In other words, it is a collection of nodes where each node has three parts i.e., link to the previous node, data which contain the elements and link to the next node.

Double linked list consumes more memory. It can be traverse in both the direction.

STRUCTURE REPRESENTATION FOR DOUBLE LINKED LIST:

```
Struct dlist
{
    Struct dlist *prev;
    Int data;
    Struct dlist *next ;};
```

3) CIRCULAR QUEUE:

Circular linked list is a collection of nodes where last node points back to the first node which means last node contains the address of first node.

CIRCULAR LINKED LIST IS CLASSIFIED INTO TWO TYPES:

- Circular single linked list
- Circular double linked list

4) HEADER LINKED LIST:

Header linked list is a linked list which contains collection of nodes along with one extra node which is called as header linked list. Header node data is used to store some information such as number of nodes in linked list, address of last nodes, etc.

Non linear list:

The data structure is the data item in which the item is not organized in the particular manner it is called as non linear data structure. A data element in the list could be connected to one or more elements in the list present in the data.

Non-linear list consists of: (a) Trees

(b) Graph.

Trees: A tree is a linked tree in which all the nodes are set in the group where each node has a value and a list of the other nodes this node are called as child nodes. There is also the main part of the two nodes as downward reference point which points to the same node.

Graphs: It is a type of data in which the data is meant to implement all the undirected graphs and directed graphs from the given nodes.

Memory is allocated to the nodes using dynamic memory allocation functions such as

Malloc (), calloc (), realloc () and free ().

1. Malloc (): This function is used to allocate a complete single block of memory of the specified size. A pointer is used to store the address returned my Malloc.

Syntax –

Data type*ptr = (data type*) malloc (size)

2. Calloc (): It is function which allocates a specified size of memory in multiple blocks of same size. Each block should be assigned to null. A pointer is used to store the address.

Syntax –

Data type*ptr = (data type*) calloc (size, number of blocks)

3. Realloc (): For reallocating the allocated memory this function is used. A pointer is used to store the address returned.

Syntax –

Data type*ptr= (data type*) realloc (ptr, size)

4. Free (): It is a function which is used to free the allocated memory.

Syntax –

Free (pointer name)

LINKED LIST

A linked list is a linear data structure, in which the elements are not sorted at contiguous memory locations, a linked list contains of nodes where each node contains a data field and a reference (link) to the next node in the list. The first node is called head. If the linked list is empty, then the value of head is NULL.

MAIN ADVANTAGES OF LINKEDLIST OVER ARRAYS IS:

1. Size of array is fixed; we must know its upper limit in advance. But in linked list size is not fixed.

2. Insertion and deletion is easy compared to array.

3. No memory wastage will be there in linked list.

The main difference between linear and non linear data structure is the data item organized data in which the data are not organized in the sequential manner this is called as non linear data structure. A data element of the non linear data structure could be connected to one or more elements present in the data.

COMMON OPERATIONS PERFORMED ON LINKED LIST:

- Inserting a node at beginning
- Inserting a node at the end
- Inserting a node at a given position
- Deleting a node at beginning
- Deleting a node at the end
- Deleting the node at a given position
- Search a node in a linked list
- Display the status of linked list
- Counting the number of nodes

A data structure is a data organization, management and storage of the data in the format that enables efficient access and modification. More precisely, a data structure is a collection of data values and systems, the relationships among them, and the functions or operations that can be applied to the data by a structure.

In my project I am using data structure of the QUEUE. In which the way which I took is in the manner of circular queue.

QUEUE:

It is a linear data structure where the first element is inserted from one end of the data which is called as REAR. And the deleted element from the other end is called FRONT .In a queue, one end is always used to insert data which is been inserted in the queue (en queue) and the other is used to delete the element from the data (de queue), because queue is open at both its ends.

Front points the beginning of the queue which is been inserted first and the Rear points to the end of the queue which is been inserted at the end.

Types of queues are:

Linear queue

Circular queue

Priority queue

Double ended queue

The main operations of the queue

En queue (): This function defines the operation for adding an element into
. Queue at starting

De queue (): This function defines the operations for removing an element from
. Queue at the ending

In it (): this function is used for initializing the queue from the main data.

Front: front is used to get the front data item from a queue to insert the elements.

Rear: rear is used to get the last item from a queue to delete the elements.

Linear queue: The linear queue takes the elements at the first and deletes the elements in the end of the queue. In which the circular queue stores all the Data in the circular fashion.

Priority queue: In the data type in which the data is like a regular queue or stack data structure. But in additional each element has its own priority. In which the An element with high priority is served before the low priority elements.

Double ended queue: This is a type of queue in the elements can be added or removed from either front or back. It is also called as head-tail linked list.

CIRCULAR QUEUE:

Circular Queue it is a linear data structure in which the data is performed based on FIFO (First in First Out) principle and the last position is connected back to the first position to make a circle in which this offers as the circular queue. It is also called 'Ring Buffer'.

The indexes 0 and 1 can only be used after the queue is reset when all the elements are placed in the data

Circular queue works by the process of the circular increment or in the circle manner.

When we try to increment any variable and we reach the end of queue we start from the beginning of the queue by the module division with the queue size is been incremented by the data structure of the queue.

It helps in avoiding the wastage of space in a regular queue implementation by array

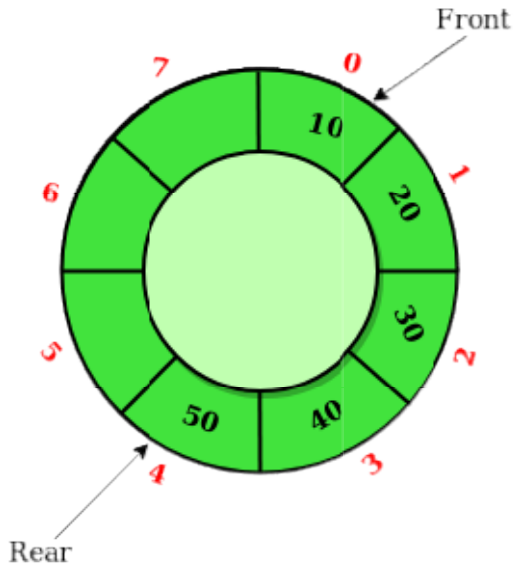

FIG 2.3 Representation of circular queue

Case 1: FRONT = 0 && REAR == SIZE – 1

Case 2: FRONT = REAR + 1

Steps

1. Check whether queue is Full – Check ((rear == SIZE-1 && front == 0) || (rear == front-1)). 2. If it is full then display Queue is full. If queue is not full then, check if (rear == SIZE – 1 && front! = 0) if it is true then set rear=0 and insert element.

3) DOUBLE ENDED QUEUE: The double ended queue is also known as dequeue. It is a queue in which insertion and deletion operations are possible at both the ends (i.e., front and rear side) of the queue.

Types of double ended queue:

- Input restricted double ended queue
- Output restricted double ended queue

INPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which insertion happens only at one end whereas deletion can only happens at both the ends.

OUTPUT RESTRICTED DOUBLE ENDED QUEUE:

It is the queue in which deletion can happen only at one end whereas insertion can happens at both the ends.

4) PRIORITY QUEUE:

Priority queue is the queue in which insertion operation and deletion operation happens based on the priority. The deletion operation is performed in accordance to priority number (i.e., data items which has highest priority is removed first from the queue) and insertion is performed only in the order.

TRAFFIC SYSTEMS :

In our modern computer controlled traffic system, it is mainly used in the circular queues that are used to switch on the traffic lights one by one repeatedly as per the time set in our signs.

Traffic management signals:

The safety, of the people as well as the smooth flow of traffic is very important in the roads. It doesn't just only make people feel secure, but it also provides safety for people. This is why it is necessary to have traffic signals that will help in organizing to the flow of traffic on the roads and people crossing the roads.

. It is necessary not only in the streets but also overall the world to have a traffic management that is well-functioning and properly maintained in the particular places. Because, as I stated earlier, it is important for the safety and security of not only drivers, but also for pedestrians crossing the road in which people be secure.

This traffic management signals all mainly used to control the traffic in order to make the people move in easy and simpler way in which the people looks after the safety of the people

We can control the traffic by the given methods:

Maintain the gap between the cars

Stay at a consistent speed

Avoid changing lanes too often

Allow other driver to merge into your lane

Pay attention to the road conditions

Avoid taking your eyes off the road

Pull over quickly and completely if you have any problem.

Importance of traffic management signals

It keeps the people safety of passengers, drivers, and pedestrians.

It keeps or makes to control of the order of the traffic movement at a particular position.

It helps in the reducing of the frequency of some type of accidents.

It tells people when to go or when to stop, thus making it beneficial in maintaining the order on the road to make safety as possible. It prevents total accidents to happen on a daily basis .Reduces the number of accidents and road issues in our locality.

Nowadays, traffic signals have become a necessary in our day-to-day life. It becomes part of our life. As well as maintaining the roads, traffic management signals are becoming used more and more on industrial sites, building sites etc, where there is regular movement of traffic.

CHAPTER 3

ALGOARITHM:

An algorithm is a step-by-step instructions or procedures for solving the problem. Algorithm forms the basic foundation of writing a program as well as to understand the program.

Before writing the programs, the following has to be known:

- What is the input required?
- Which are the tasks to be performed?
- What should be the output of the program?

- Step 1: start the program

- Step 2: insert and declare all header files which are required for this project

- Step 3: and declare all variables which were required for project

- Step4: create the structure for the link and create a node by using circular queue user can display the traffic signals according to

- RED -> YELLOW -> GREEN ->RED-> YELLOW.....

- Step 5: Next we use display function for displaying the colors of the signals.

- Step 6: after that the user will create an main function which consists of switch case and user declare an two options in switch case with the help of the switch case user will display all the three functions.

- Step7: then user use an default statement to display an wrong choice

- Step 8: exit/quit from the code.

Chapter 4:

IMPLEMENTATION/MODULE FUNCTIONALITY

4.1 MODULE 1

Insert: Inserts an element to the data structure if it is not already present. If not present then insert an element at the end of the array. If added as the key and last array index as the index and prints the array.

*head implies that you're passing a pointer struct node is the type of the pointer you're passing to the data. In this case it is a structure; the structure is defined as a node of the pointer. Therefore we are passing a pointer which points to an object of the structure given in the data.

Insert (name of the pointer to object);

4.2 MODULE 2

Void display (): display function is used to print the values at a time .Void display is not a function which displays the elements in the present order.

This is also called as prototyping of the given argument list. In which we can show the elements which are been displayed. We can display all the elements in which we have inserted in the data.

Display (inserted elements)

Chapter 5:

Result:

fig 5.1 shows to enter the first signal colour which is to be there

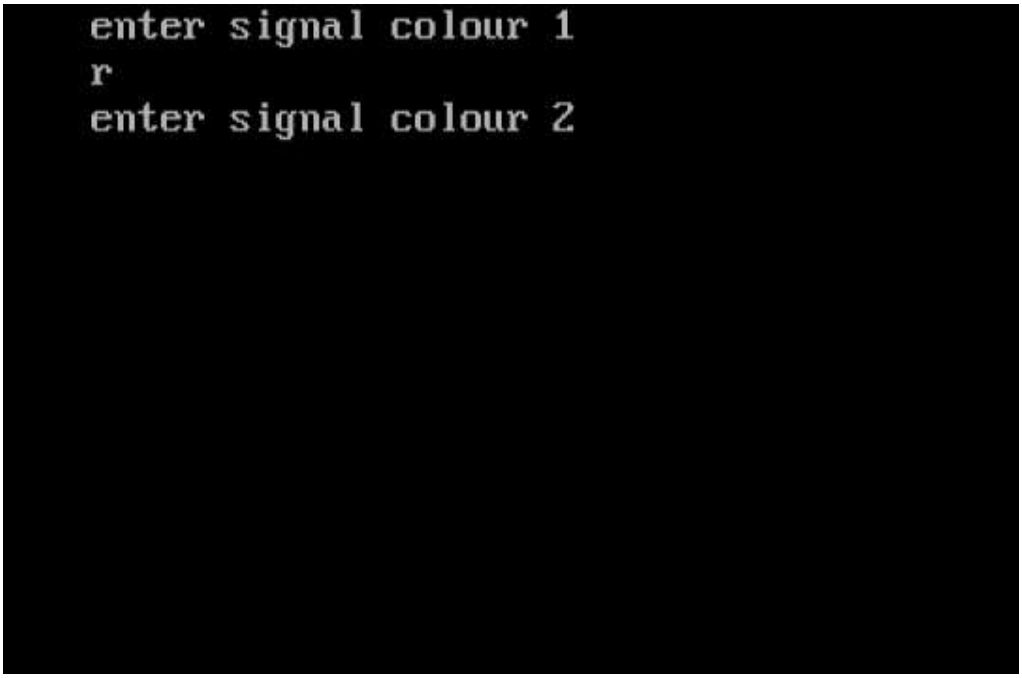

Fig 5.2 shows that when we have entered the first colour and then to enter the second colour

Fig 5.3 shows that when we have entered the first colour and then to enter the second colour and now to enter the third colour

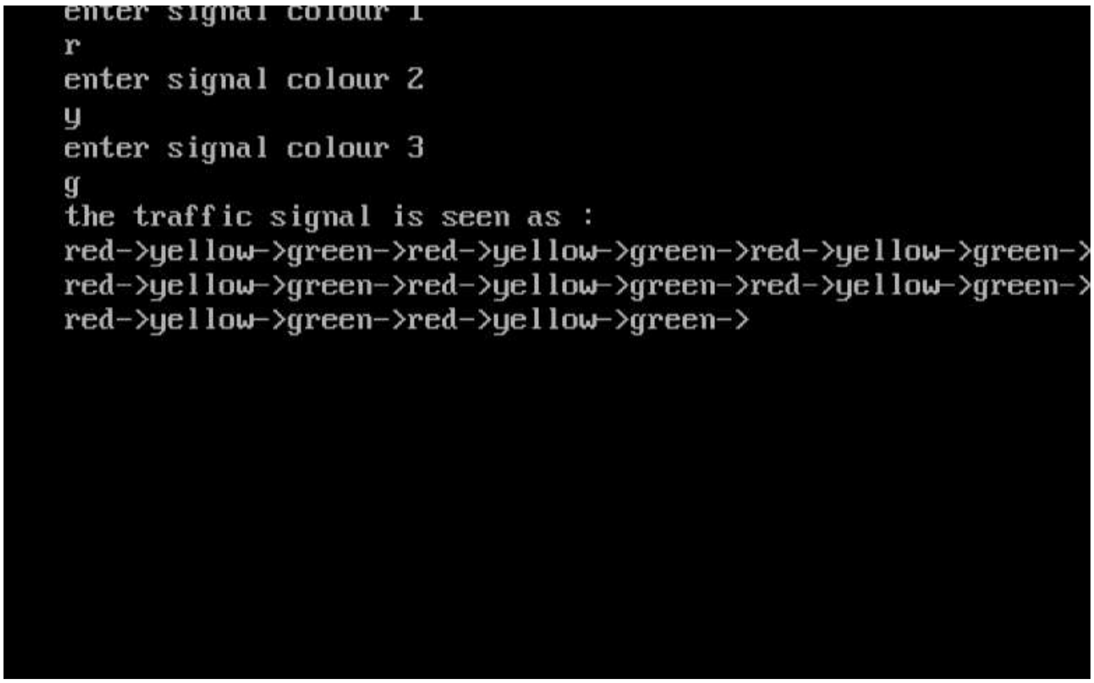

Fig 5.4 shows that we have entered all the colors and then it prints the colors to our given range

```
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->yellow->green->red->yellow->green->red->yellow->green->
red->yellow->green->red->
```

Fig 5.5 shows that if we want it as the circular queue till we quit then we get it

Chapter 6:

Conclusion:

The traffic signals are set to provide the green light on demand or only in the presence of the vehicles and to move the vehicles. As a conclusion, the controller can control the traffic movements and deletes a busy and non busy road by changing the signals. The overall of the project is ok but certain condition the traffic signals will not function properly

I thank my reviewer Ms. Heyshanthini for helping me to complete this project. I have learned many things about data structures especially linked list through this project

REFERENCE

1. https://www.geeksforgeeks.org
2. https://www.wikipedia.org
3. https://developerinsider.co/download-turbo-c-for-windows-7-8-8-1-and-windows-10-32-64-bit-full-screen/
4. https://circularqueue.geeksforgeeks.com