DSA

DATA STRUCTURE SIMULATOR

An interactive learning tool

Submitted by

Habibur Rahman Mahin

BSSE-1422

Institute of Information Technology

University of Dhaka

Supervised by

Dr.Ahmedul Kabir

Associate Professor

Institute of Information Technology

University of Dhaka

About DSS

The **Data structure Simulator**(**DSS**) is designed to provide a visual and interactive learning experience for users to understand various data structures and sorting algorithms.The simulator aims to help beginners and students gasp the underlying concepts of fundemental data structures like stacks,queues,trees,graphs sorting techniques such as bubble sort,insertin sort,merge sort etc.

Goals

Objective 01

To make data structures easier to understand for the users

Objective 02

To let users visualize how various algorithms actually work

Project Features

01

Interactive Interface

02

Step-by-Step Process

03

Dynamic Data Input

04

Interactive Manipulation

Interactive Interface

The simulator boasts a user-friendly interface,allowing users to selectt and interact with different data structures and algorithms easily.Users can visualize how data elements are stores,accessed and manipulated step-by-step.

Step-by-Step Process

The primary objective of the Data Structure Simulator is to provide a step-by-step demonstration of how each data structure and sorting algorithm works.Users can follow along with the simulation to undersatnd the data flow and changes at each stage of the operations

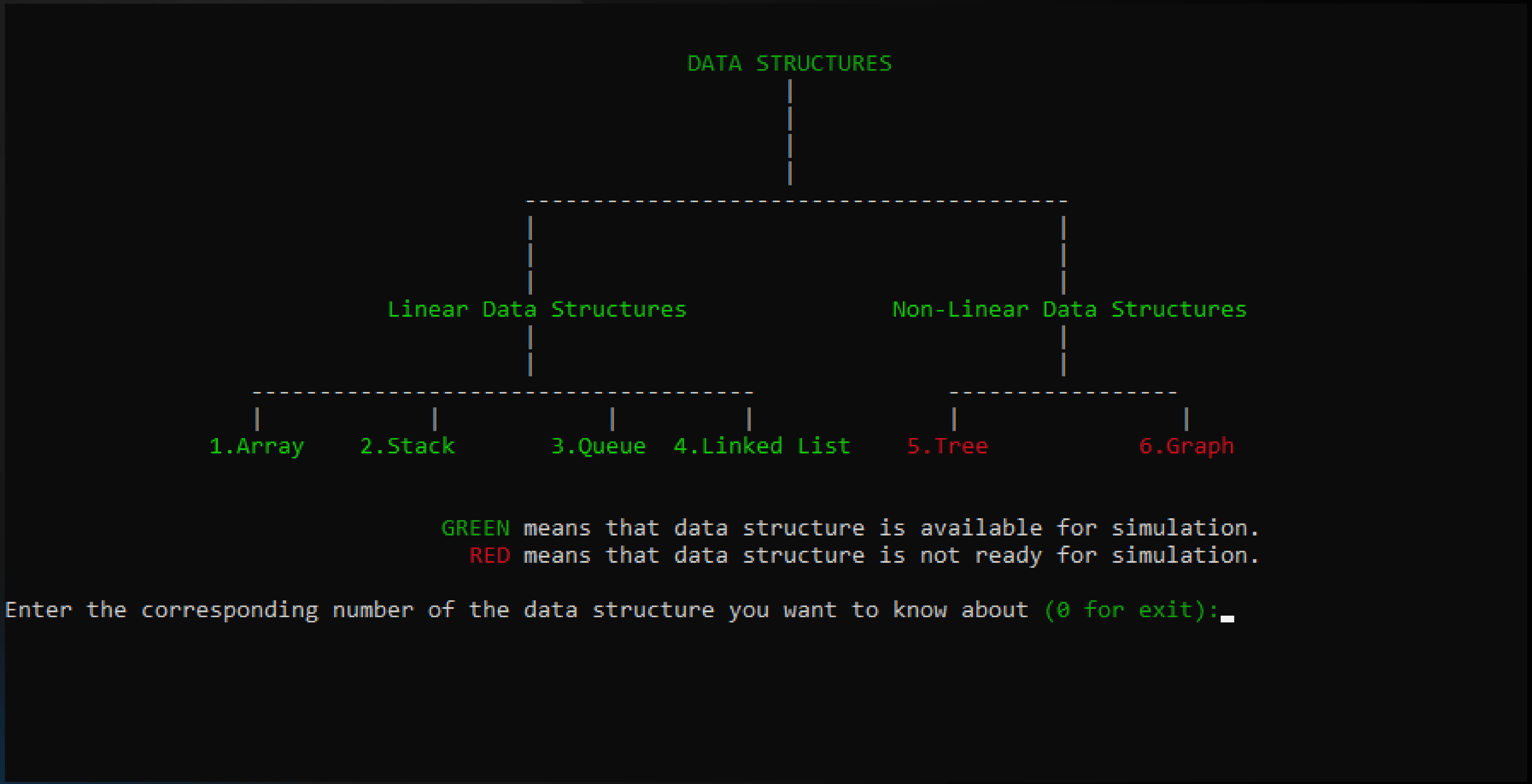
Dynamic Data Input

Users have the option to input their data elements to the data structures and sorting algorithms.This flexibility allows them to see firsthand how the structure handles different data sets and how sorting algorithms arrange them in real-time.

Interactive Manipulation

DSS allows users to interact with the simulated structure directly.They can add,remove,or modify elements ,test various operations,and observe how these actions affect the overall structure.

User-friendly interface

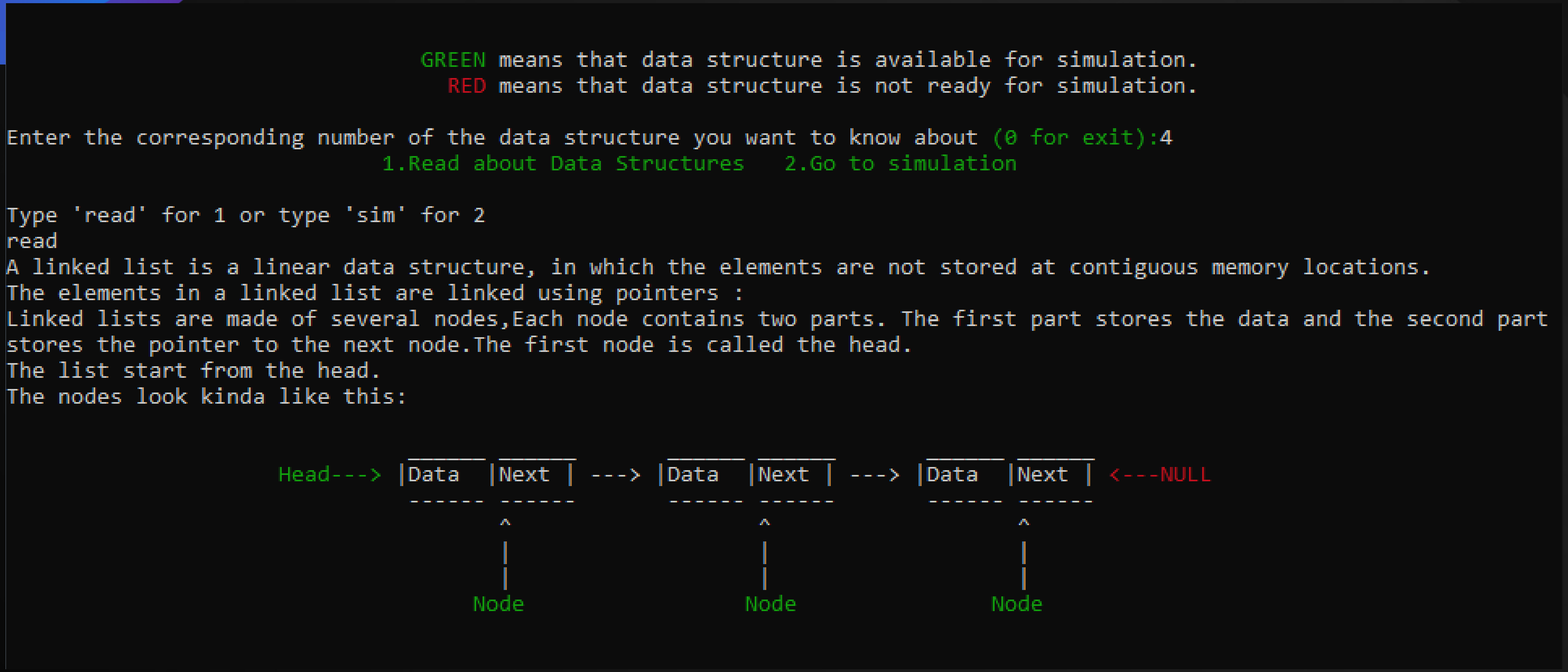


```
                                                  DATA STRUCTURES
                                                         |
                                                         |
                                                         |
                                                         |
                                      ----------------------------------------
                                      |                                      |
                                      |                                      |
                                      |                                      |
                            Linear Data Structures               Non-Linear Data Structures
                                      |                                      |
                                      |                                      |
                  -------------------------------------              -----------------
                  |            |            |         |              |                |
               1.Array    2.Stack       3.Queue  4.Linked List    5.Tree           6.Graph


                                GREEN means that data structure is available for simulation.
                                  RED means that data structure is not ready for simulation.

Enter the corresponding number of the data structure you want to know about (0 for exit):
```

Get visually adapted to different data structures

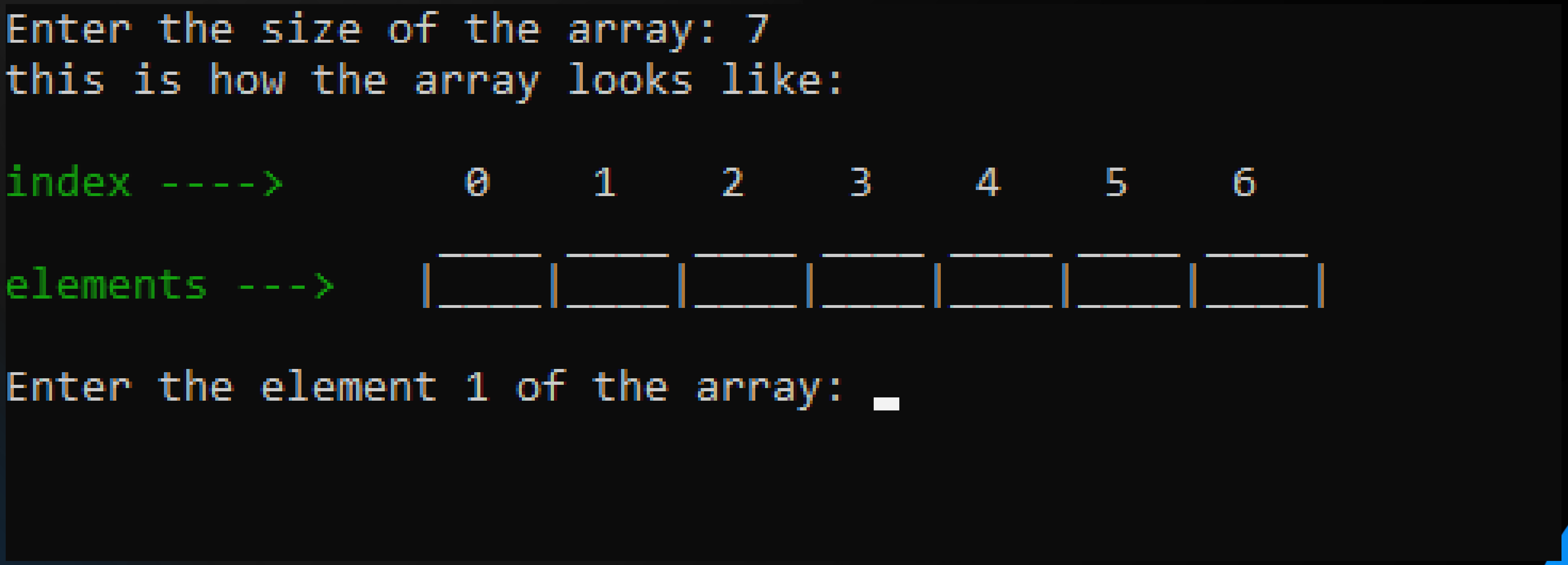


```
                                GREEN means that data structure is available for simulation.
                                  RED means that data structure is not ready for simulation.

Enter the corresponding number of the data structure you want to know about (0 for exit):4
                             1.Read about Data Structures   2.Go to simulation

Type 'read' for 1 or type 'sim' for 2
read
A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations.
The elements in a linked list are linked using pointers :
Linked lists are made of several nodes,Each node contains two parts. The first part stores the data and the second part
stores the pointer to the next node.The first node is called the head.
The list start from the head.
The nodes look kinda like this:


                     ______ ______       ______ ______       ______ ______
         Head---> |Data   |Next |  ---> |Data   |Next |  ---> |Data   |Next |  <---NULL
                   ------ ------       ------ ------       ------ ------
                          ^                   ^                   ^
                          |                   |                   |
                          |                   |                   |
                         Node                Node                Node
```

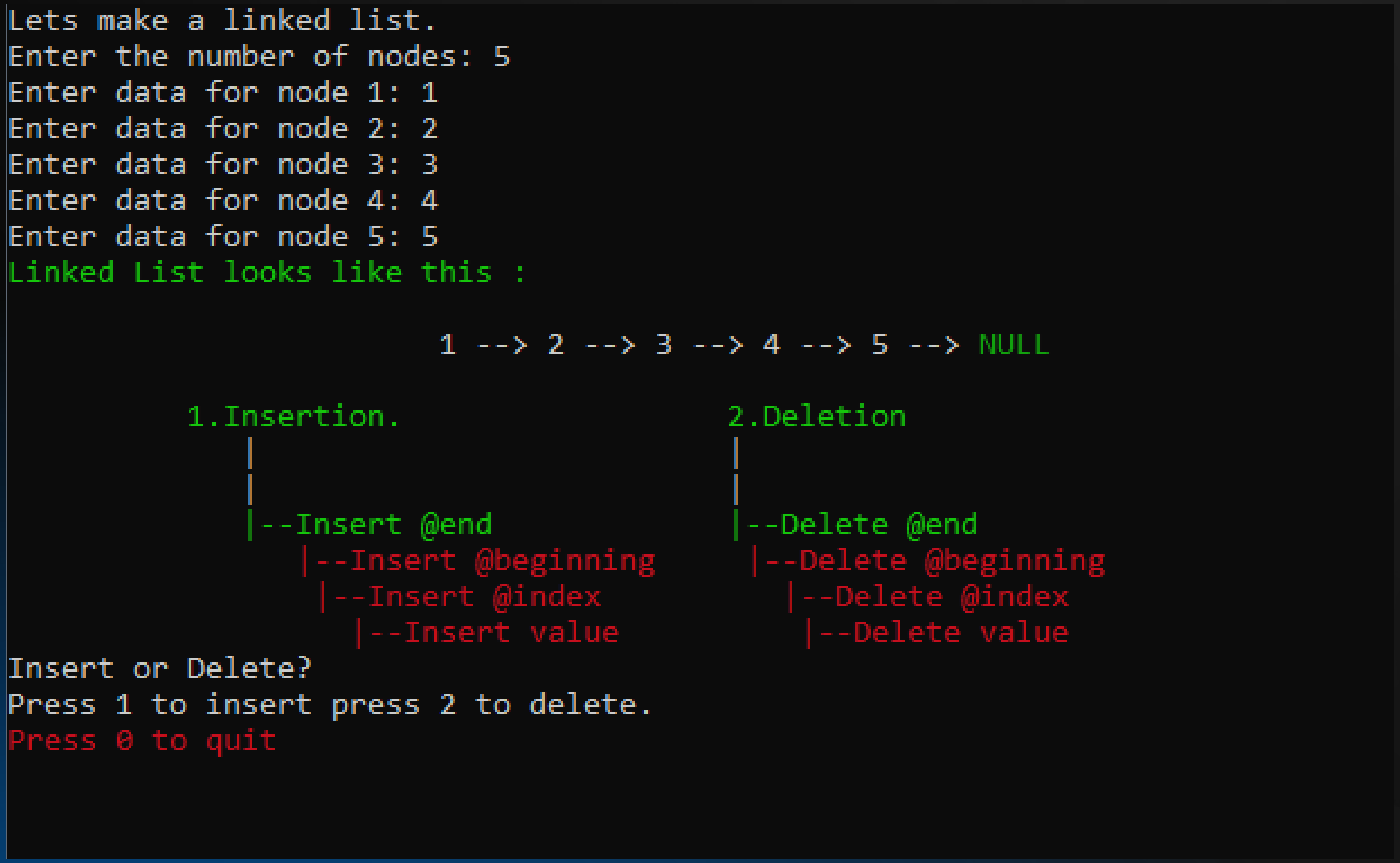


```
Enter the size of the array: 7
this is how the array looks like:

index ---->        0    1    2    3    4    5    6
                 ____ ____ ____ ____ ____ ____ ____
elements --->   |____|____|____|____|____|____|____|

Enter the element 1 of the array: _
```

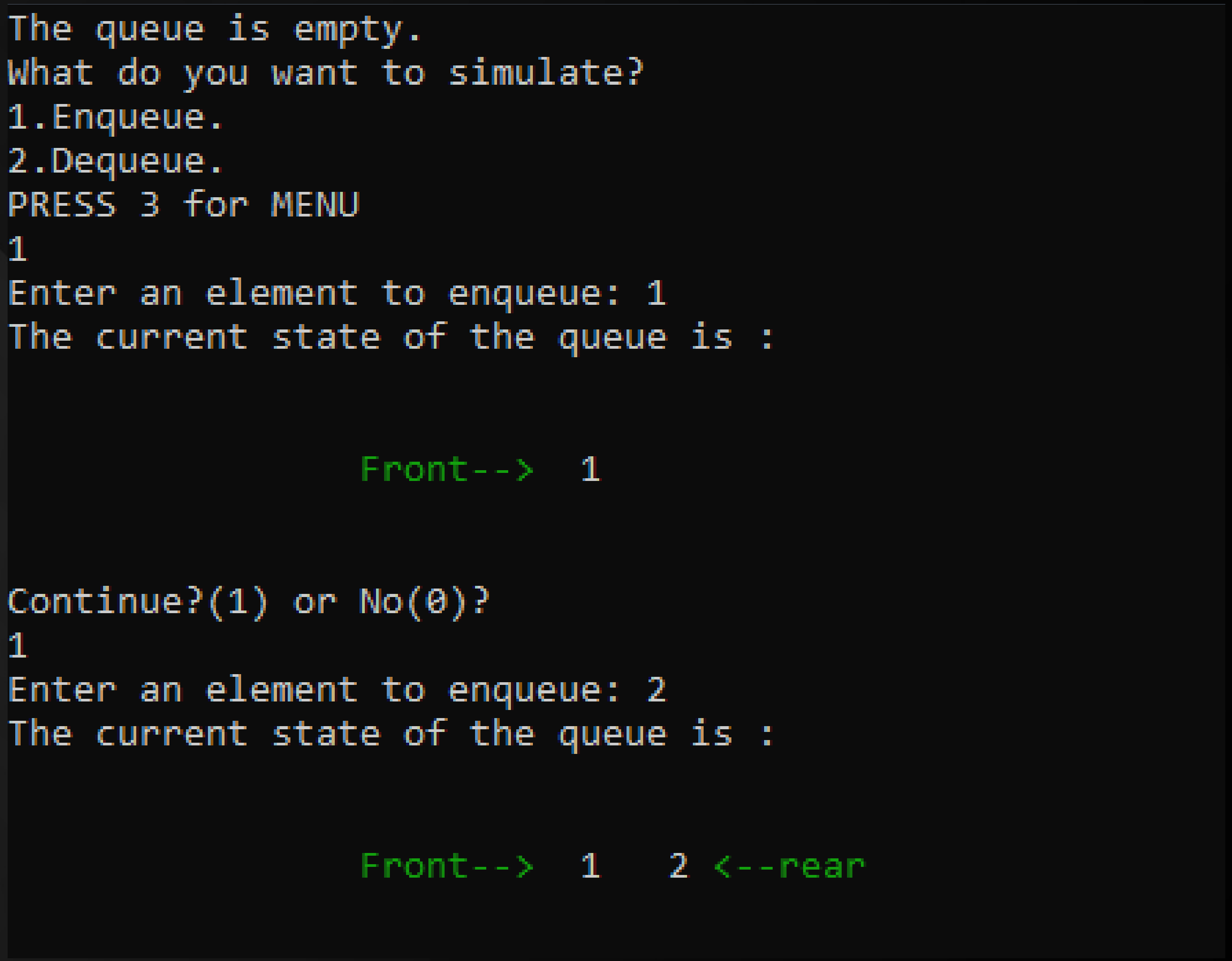
Dynamic inputs and outputs



```
Lets make a linked list.
Enter the number of nodes: 5
Enter data for node 1: 1
Enter data for node 2: 2
Enter data for node 3: 3
Enter data for node 4: 4
Enter data for node 5: 5
Linked List looks like this :

                        1 --> 2 --> 3 --> 4 --> 5 --> NULL

          1.Insertion.                  2.Deletion
             |                          |
             |                          |
             |--Insert @end             |--Delete @end
                |--Insert @beginning     |--Delete @beginning
                 |--Insert @index          |--Delete @index
                   |--Insert value          |--Delete value
Insert or Delete?
Press 1 to insert press 2 to delete.
Press 0 to quit
```



```
The queue is empty.
What do you want to simulate?
1.Enqueue.
2.Dequeue.
PRESS 3 for MENU
1
Enter an element to enqueue: 1
The current state of the queue is :


                Front-->  1


Continue?(1) or No(0)?
1
Enter an element to enqueue: 2
The current state of the queue is :


                Front-->  1   2 <--rear
```

DSA

THANK YOU

For watching this presentation