National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Data Structures**	**Course Code:**	**CS 218**
Program:	**BS (DS)**	**Semester:**	**Fall 2024**
Due Date:	**17th of September, 2024.**	**Total Marks:**	**120**
Section:	**3A**	**Page(s):**	**5**
Type:	**Assignment 1**		

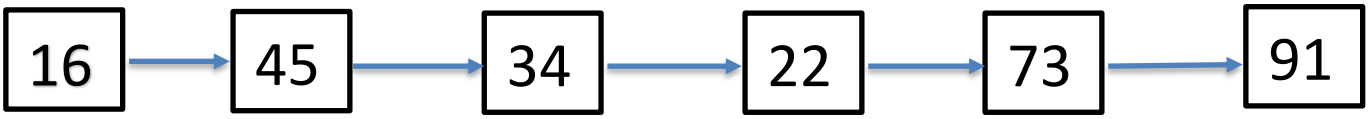
Q: Determine the time complexity of the following code snippets. (60)

Viva will be conducted of similar examples during evaluation.

int Sum=0 **for(int i=1; i<=N; i++)** **for(int j=1; j<=N; j++)** **Sum++;**	**2)** **int Sum=0;** **for(int i=1; i<N; i*=2)** **for(int j=1; j<=i; j++)** **Sum++;**
3)What is the algorithm's complexity of the following piece of code. **int Sum=0;** **for(int i=1; i<=N; i++)** **for(int j=1; j<=N; j++)** **for(int k=1; k<=N; k++)** **Sum++;**	**4)** **int Sum=0;** **for(int i=1; i<=N; i++)** **for(int j=1; j<=i; j++)** **for(int k=1; k<=j; k++)** **Sum++;**
5) **int Sum=0;** **for(int i=1; i<N; i*=2)** **for(int j=1; j<N; j*=2)** **Sum++;**	**6)** **Int Sum=0;** **for(int i=1; i<=N*N; i+=2)** **for(int j=1; j<N*N; j*=2)** **Sum++;**
7) **int Sum=0;** **for(int i=1; i<=N; i++)** **for(int j=1; j<=i*i; j++)** **Sum++;**	**8)** **int Sum=0;** **for(int i=1; i<N*N; i*=2)** **for(int j=1; j<N*N; j*=2)** **Sum++;**
9) **int Sum=0;** **for(int i=1; i<=N; i++)** **for(int j=1; j<=i; j++)** **for(int k=1; k<=j; k++)** **Sum++;**	**10)** **int Sum=0;** **for(int i=1; i<=N; i*=2)** **for(int j=1; j<i; j*=2)** **Sum++;**
11) **int Sum=0;** **for(int i=1; i<=N*N; i*=2)** **Sum++**	**12)** **int Sum=0;** **for(int i=1; i<=sqrt(N); i*=2)** **for(int j=1; j<=1000000; j++)** **Sum++;**

Question No. 2: (20 Marks)

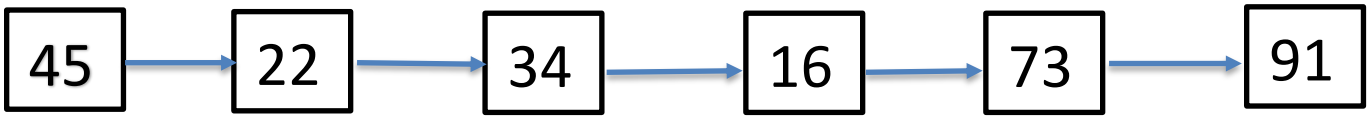
Make a linked list class with all required functionalities.
Keep taking integer inputs from the user unless -1 is entered.
The inputs are the indexes of the linked list you need to arrange in DECENDING order.



Take the linked list above as an example, if the user enters the following values:

4
2
1
-1

The linked list should be altered as following:



You are not allowed to swap the data. The nodes should be swapped.
Creation/Deletion of nodes or linked list is strictly not allowed.

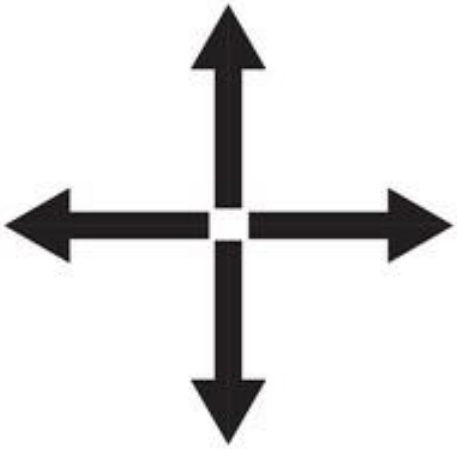
NOTE:

The user can enter as many values as much they want. If some index entered by user is an invalid index, ignore that value and word on the valid indexes, and also display that which particular index was invalid.

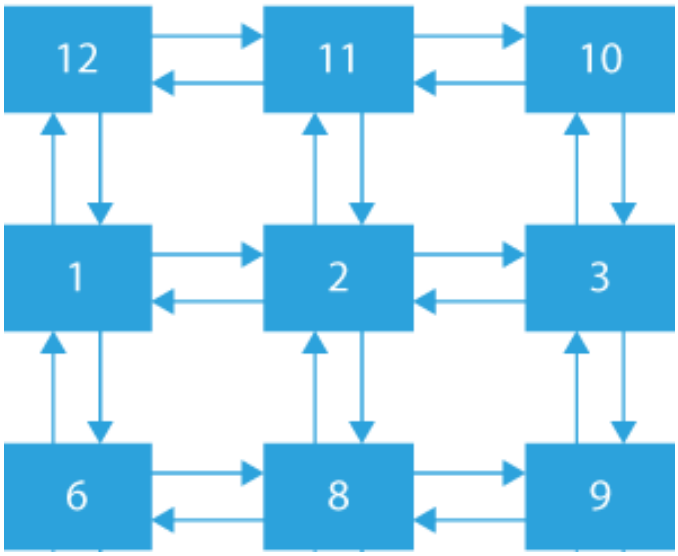
Question No. 3 (40 Marks)

Luck is the second most important thing to reach what you're looking for. Hard work and following the clues is the first. Your task is to reach the elite node in the network of linked lists.

Each node will have following members:



The entire structure will look like:



You have to make the maze using file, where each line represents a row with values seperated by commas. The file of the figure will look like :

12,11,10
1,2,3
6,8,9

You wil start from the top left corner node and reach the next node using the data as the Clue.
Decode the clue as following:

Row of next node = (sum of all digits % No. of rows) + 1
Column of next node = number of digits

Elite Node: The node whose clue will bring you to itself will be the elite node.

Functionalities:

- ✓ **Read(string filename)**

  This method should be able to read data from a text file. Each line in the file contains all the entries in that row, values seperated by commas. There is one row per line.

  12,34,56,78,90
  44,76,34,87,99
  88,65,12,19,50

  This data shows that the maze will consist of 3 rows and 5 columns.

- ✓ **Print(node* head)**

  Prints the maze created.

- ✓ **ClueRow(int data)**

  Returns the row number of the next node to be visited (1st row is called row 1).

- ✓ **ClueColumn(int data)**

  Returns the column number of the next node to be visited (1st column is called column 1).

✓ **Visited(node* Current)**

Prints the data of the node you're currently at.

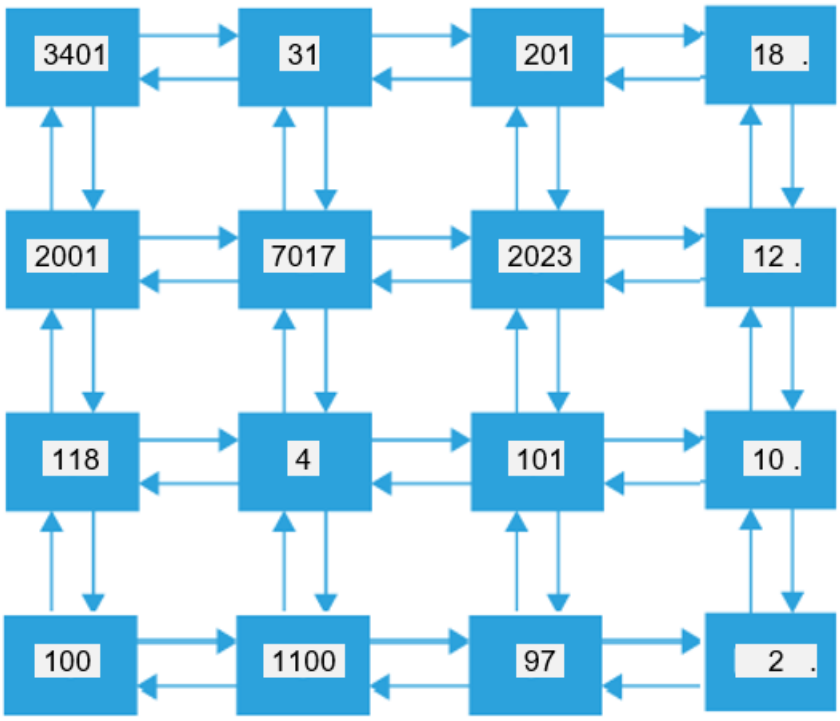
You need to print all the nodes you visit while reaching the elite node.

✓ **EliteNode(node* elite)**

When elite node is found, send it to this function to print the data and a message that elite node has been found.

If the there doesn't exist any elite node, your program should be able to display a message for it. Also, if any node get's visited for the second time, the program should end after displaying a meaningful message.

Input:



Output: 3401->18->7017->2->118->101
Elite Node = 101

YOU ARE NOT ALLOWED TO MAKE A 2D MATRIX AT ANY INSTANCE IN YOUR PROGRAM. READING DATA THROUGH FILE AFTER THE CREATION OF THE MAZE IS ALSO NOT ALLOWED.
Traverse the maze (left / right / top / bottom) to reach the next node.