Bahria University,

Karachi Campus



COURSE: CSC-221 DATA STRUCTURES AND ALGORITHM

TERM: FALL 2020, CLASS: BSE- 3 (A)

Submitted By:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ADIL WAHEED) (65190)

ENROLLMENT NO #:02-131192-082

Submitted To:

Engr. Maam Nazar Mobeen/ Engr. Ramshaa

Signed Remarks: Score:

INDEX

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
| 01 | 1-10-2020 | 01 | ONE AND TWO DIMENSIONAL ARRAY |  |
| 02 | 09-10-20 | 02 | Linear Search & Sorting Algorithms |  |
| 03 | 13-10-20 | 03 | Recusrion |  |
| 04 | 30/10/2020 | 04 | Binary Search Algorithm |  |
| 05 | 30/10/2020 | 05 | Merge Sort |  |
| 06 | 30/10/2020 | 06 | Quick Sort |  |
| 07 | 4/11/2020 | 07 | Stack |  |
| 08 | 12/11/2020 | 08 | QUEUE |  |
| 09 | 20/10/2020 | 09 | Doubly Linked List |  |
| 10 | 24/10/2020 | 10 | CIRCULAR Linked List |  |
| 11 | 1/1/2021 | 11 | BUCKET, RADIX SORT & BINARY TREES |  |
| 12 | 4/1/2021 | 12 | BST AND AVL TREE |  |
| 13 | 4/1/2021 | 13 | **G**RAPHS |  |
| SNO | DATE | LAB NO | LAB OBJECTIVE | SIGN |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_13\_\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | **Create a program to implement Graphs With Adjacency matrix** |
| 2 | **Create a program to implement Graphs With Adjacency list** |
|  |  |
|  |  |
|  |  |

Submitted On:

\_\_\_\_\_\_\_\_\_\_\_\_

(Date: 4/1/2021)

**Task No. 1:**

**Create a program to implement Graphs With Adjacency matrix**

Solution:

Console.WriteLine("Enter the number of Nodes : ");

int n = int.Parse(Console.ReadLine());

string[] arr = new string[n];

for (int i = 0; i < n; i++)

{

Console.WriteLine((i + 1) + " is connected to nodes?");

arr[i] = Console.ReadLine();

}

int[,] array = new int[n, n];

for (int i = 0; i < array.GetLength(0); i++)

{

for (int j = 0; j < array.GetLength(1); j++)

{

string temp = "" + (j + 1);

if (arr[i].Contains(temp))

{

array[i, j] = 1;

}

else

{

array[i, j] = 0;

}

}

}

Console.Write(" ");

for (int i = 0; i < n; i++)

{

Console.Write(i + 1 + " ");

}

Console.WriteLine();

for (int i = 0; i < array.GetLength(0); i++)

{

Console.Write((i + 1) + " ");

for (int j = 0; j < array.GetLength(1); j++)

{

Console.Write(array[i, j] + " ");

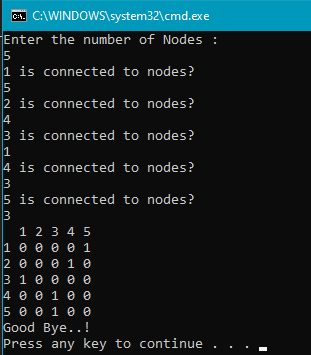
}

Console.WriteLine();

}

Console.WriteLine("Good Bye..!");

**OUTPUT:**

****

**Task No. 2:**

**Create a program to implement Graphs With Adjacency list**

Solution:

Console.WriteLine("Enter the number of nodes");

int n = int.Parse(Console.ReadLine());

string[] arr = new string[n];

for (int i = 0; i < n; i++)

{

Console.WriteLine((i + 1) + " is connected to nodes?");

arr[i] = Console.ReadLine();

}

List<int>[] obj = new List<int>[n];

for (int i = 0; i < n; i++)

{

obj[i] = new List<int>();

string[] temp = arr[i].Split(',');

for (int a = 0; a < temp.Length; a++)

{

obj[i].Add(int.Parse(temp[a]));

}

}

for (int i = 0; i < n; i++)

{

Console.Write((i + 1) + " => ");

for (int l = 0; l < obj[i].Count; l++)

{

Console.Write(obj[i][l] + " ");

}

Console.WriteLine();

}

Console.WriteLine("Good Bye...!");

**OUTPUT:**

