Digital Assignment – 4

Data Structures

Name: Hari Krishna Shah

VIT ID: 21BCS0167

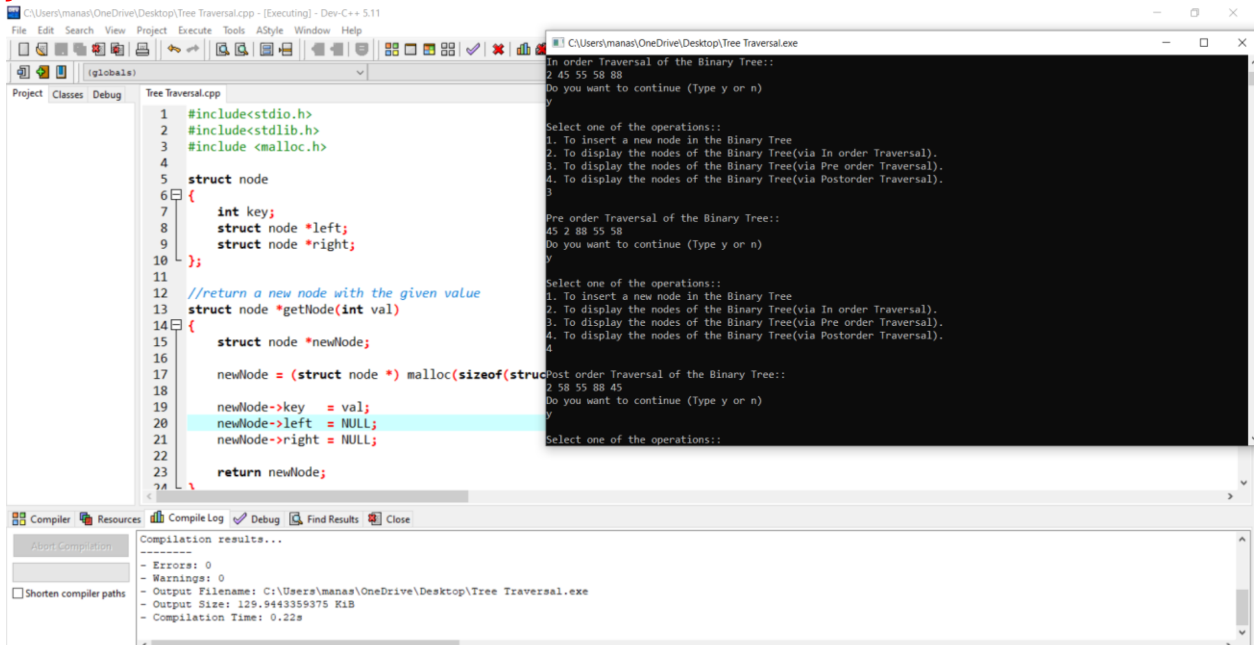
Link:
https://drive.google.com/drive/folders/1PUqW45L5odXgHRW0SZnIwJMwWzt35bLq?usp=sharing

Ques 1. Implement tree traversal program Inorder , pre order and post order.

Answer:

```
#include<stdio.h>
#include<stdlib.h>
#include <malloc.h>

struct node
{
    int key;
    struct node *left;
    struct node *right;
};

//return a new node with the given value
struct node *getNode(int val)
{
    struct node *newNode;

    newNode = (struct node *) malloc(sizeof(struct node));

    newNode->key   = val;
    newNode->left  = NULL;
    newNode->right = NULL;

    return newNode;
}

//inserts nodes in the binary search tree
struct node *insertNode(struct node *root, int val)
{
     if(root == NULL)
         return getNode(val);

     if(root->key < val)
         root->right = insertNode(root->right,val);

     if(root->key > val)
         root->left = insertNode(root->left,val);

     return root;
}

//inorder traversal of the binary search tree
void inorder(struct node *root)
{
    if(root == NULL)
        return;

    //traverse the left subtree
    inorder(root->left);

    //visit the root
    printf("%d ",root->key);

    //traverse the right subtree
    inorder(root->right);
}
//preorder traversal of the binary search tree
void preorder(struct node *root)
{
    if(root == NULL)
        return;

    //visit the root
    printf("%d ",root->key);

    //traverse the left subtree
    preorder(root->left);

    //traverse the right subtree
    preorder(root->right);
}
//postorder traversal of the binary search tree
void postorder(struct node *root)
{
    if(root == NULL)
        return;

    //traverse the left subtree
    postorder(root->left);

    //traverse the right subtree
    postorder(root->right);

    //visit the root
    printf("%d ",root->key);
}
/*
...- / ..- -. .. .-. / ... -. -.-- -.-- .-. .- / ...- .- / -.-- -
... .. .-. / .--- ...- --. ..- / .-.. -... .... / .--. ..- ...- -
-. . -. / .-. .. .-. . / ..-. ...- .- .--. .-. / ...- / --.. .-.
--. / .-.. -... .... .-.-.- / ...- / .--- ...- -.-- -.-- / -. -.-
- .--- -. .-.. ..-. / -..- .-. .-. -.-. / -.-- -... .. ...- .- -
/ .-.. -... .... .-.-.- / -.-. -.-- .-. -. ..-. .-. / --- .-. /
.--- ...- --. ..- / --.. .-. .-.-.- / .-.. -... .... . ..-. / ..-
-. . ...- / ..-. ..- -. ..- .-.-.-
*/


int main()
{
   struct node *root = NULL;


    int data;
    char ch;
```

```
        /*  Do while loop to display various options to select 
from to decide the input  */
        do
        {
            printf("\nSelect one of the operations::");
            printf("\n1. To insert a new node in the Binary 
Tree");
            printf("\n2. To display the nodes of the Binary 
Tree(via In order Traversal).");
               printf("\n3. To display the nodes of the Binary 
Tree(via Pre order Traversal).");
               printf("\n4. To display the nodes of the Binary 
Tree(via Postorder Traversal).\n");

            int choice;
            scanf("%d",&choice);
            switch (choice)
            {
            case 1 :
                printf("\nEnter the value to be inserted\n");
                scanf("%d",&data);
                root = insertNode(root,data);
                break;
            case 2 :
                printf("\nIn order Traversal of the Binary 
Tree::\n");

                inorder(root);
                break;
               case 3 :
                printf("\nPre order Traversal of the Binary 
Tree::\n");
                    preorder(root);
                break;
               case 4 :
                printf("\nPost order Traversal of the Binary 
Tree::\n");
                postorder(root);
                break;
            default :
                printf("Wrong Entry\n");
                break;
            }
            printf("\nDo you want to continue (Type y or n)\n");
            scanf(" %c",&ch);
        } while (ch == 'Y'|| ch == 'y');

    return 0;
}
```


C:\Users\manas\OneDrive\Desktop\Tree Traversal.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
Tree Traversal.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
C:\Users\manas\OneDrive\Desktop\Tree Traversal.exe

```
#include<stdio.h>
#include<stdlib.h>
#include <malloc.h>

struct node
{
    int key;
    struct node *left;
    struct node *right;
};

//return a new node with the given value
struct node *getNode(int val)
{
    struct node *newNode;

    newNode = (struct node *) malloc(sizeof(struc

    newNode->key   = val;
    newNode->left  = NULL;
    newNode->right = NULL;

    return newNode;
}
```

```
In order Traversal of the Binary Tree::
2 45 55 58 88
Do you want to continue (Type y or n)
y

Select one of the operations::
1. To insert a new node in the Binary Tree
2. To display the nodes of the Binary Tree(via In order Traversal).
3. To display the nodes of the Binary Tree(via Pre order Traversal).
4. To display the nodes of the Binary Tree(via Postorder Traversal).
3

Pre order Traversal of the Binary Tree::
45 2 88 55 58
Do you want to continue (Type y or n)
y

Select one of the operations::
1. To insert a new node in the Binary Tree
2. To display the nodes of the Binary Tree(via In order Traversal).
3. To display the nodes of the Binary Tree(via Pre order Traversal).
4. To display the nodes of the Binary Tree(via Postorder Traversal).
4

Post order Traversal of the Binary Tree::
2 58 55 88 45
Do you want to continue (Type y or n)
y

Select one of the operations::
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\manas\OneDrive\Desktop\Tree Traversal.exe
- Output Size: 129.9443359375 KiB
- Compilation Time: 0.22s
```

Ques 2. Implement graph traversal using BFS and DF.

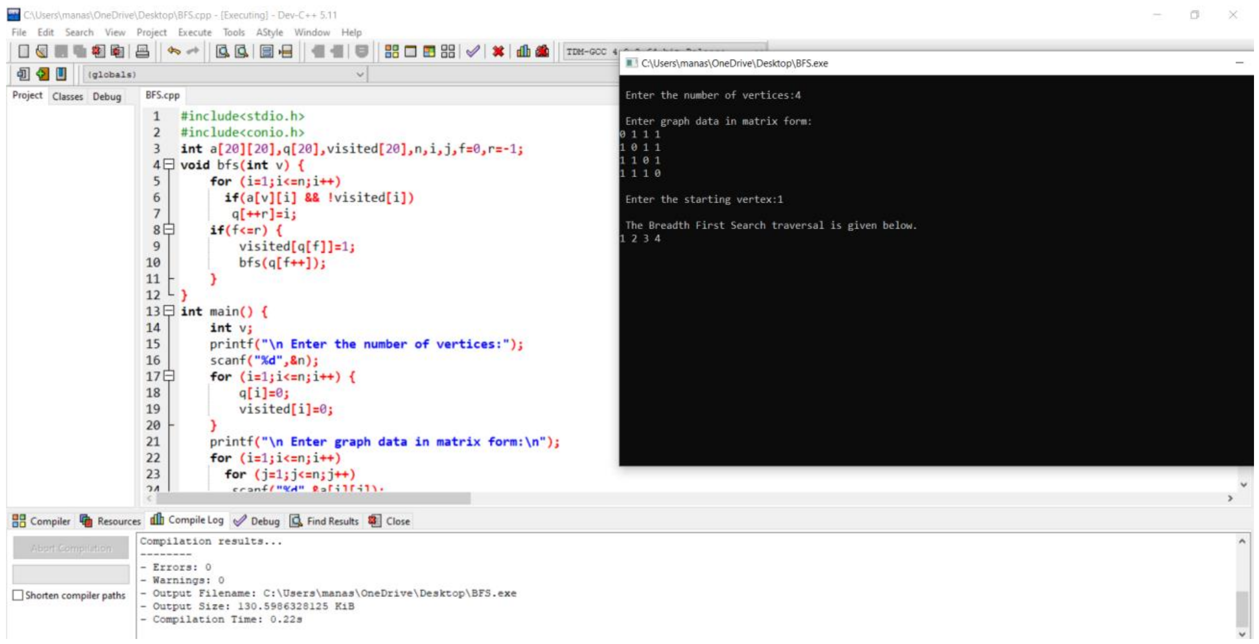
Answer:

1. **BFS**

Answer:

```
#include<stdio.h>
#include<conio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v) {
     for (i=1;i<=n;i++)
       if(a[v][i] && !visited[i])
        q[++r]=i;
     if(f<=r) {
          visited[q[f]]=1;
          bfs(q[f++]);
     }
}
```

```c
int main() {
	int v;
	printf("\n Enter the number of vertices:");
	scanf("%d",&n);
	for (i=1;i<=n;i++) {
		q[i]=0;
		visited[i]=0;
	}
	printf("\n Enter graph data in matrix form:\n");
	for (i=1;i<=n;i++)
	  for (j=1;j<=n;j++)
	   scanf("%d",&a[i][j]);
	printf("\n Enter the starting vertex:");
	scanf("%d",&v);
	bfs(v);
	printf("\n The Breadth First Search traversal is given
below.\n");
	for (i=1;i<=n;i++)
	  if(visited[i])
	   printf("%d ",i);
	else
	   printf("\n Bfs is not possible");
	getch();
}
```


C:\Users\manas\OneDrive\Desktop\BFS.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
(globals)
Project Classes Debug
BFS.cpp
C:\Users\manas\OneDrive\Desktop\BFS.exe
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths

```c
#include<stdio.h>
#include<conio.h>
int a[20][20],q[20],visited[20],n,i,j,f=0,r=-1;
void bfs(int v) {
	for (i=1;i<=n;i++)
	  if(a[v][i] && !visited[i])
	   q[++r]=i;
	if(f<=r) {
		visited[q[f]]=1;
		bfs(q[f++]);
	}
}
int main() {
	int v;
	printf("\n Enter the number of vertices:");
	scanf("%d",&n);
	for (i=1;i<=n;i++) {
		q[i]=0;
		visited[i]=0;
	}
	printf("\n Enter graph data in matrix form:\n");
	for (i=1;i<=n;i++)
	  for (j=1;j<=n;j++)
```

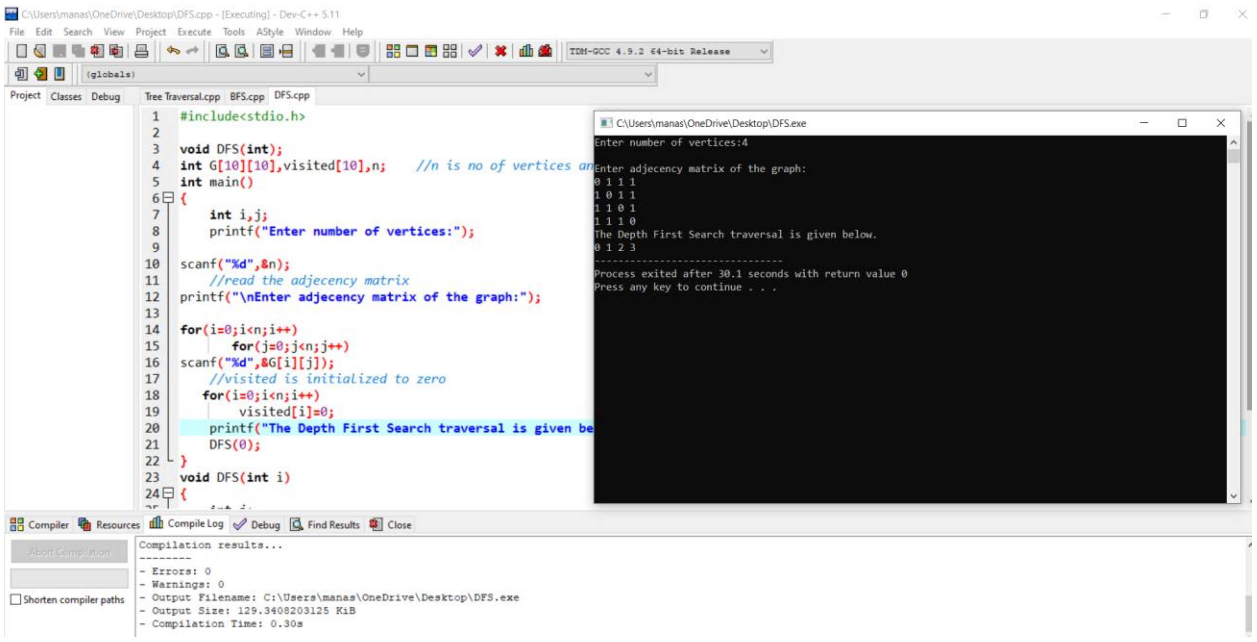
```
 Enter the number of vertices:4

 Enter graph data in matrix form:
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0

 Enter the starting vertex:1

 The Breadth First Search traversal is given below.
1 2 3 4
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\manas\OneDrive\Desktop\BFS.exe
- Output Size: 130.5986328125 KiB
- Compilation Time: 0.22s
```

2.DFS

Answer:

```
#include<stdio.h>

void DFS(int);
int G[10][10],visited[10],n;    //n is no of vertices and graph
is sorted in array G[10][10]
int main()
{
    int i,j;
    printf("Enter number of vertices:");

scanf("%d",&n);
    //read the adjecency matrix
printf("\nEnter adjecency matrix of the graph:");

for(i=0;i<n;i++)
       for(j=0;j<n;j++)
scanf("%d",&G[i][j]);
    //visited is initialized to zero
   for(i=0;i<n;i++)
        visited[i]=0;
     printf("The Depth First Search traversal is given
below.\n");
    DFS(0);
}
void DFS(int i)
{
    int j;
printf("%d ",i);
    visited[i]=1;
for(j=0;j<n;j++)
       if(!visited[j]&&G[i][j]==1)
           DFS(j);
}
```


C:\Users\manas\OneDrive\Desktop\DFS.cpp - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
Tree Traversal.cpp BFS.cpp DFS.cpp
Compiler Resources Compile Log Debug Find Results Close
Abort Compilation
Shorten compiler paths
C:\Users\manas\OneDrive\Desktop\DFS.exe

```c
#include<stdio.h>

void DFS(int);
int G[10][10],visited[10],n;    //n is no of vertices an
int main()
{
    int i,j;
    printf("Enter number of vertices:");

scanf("%d",&n);
    //read the adjecency matrix
printf("\nEnter adjecency matrix of the graph:");

for(i=0;i<n;i++)
       for(j=0;j<n;j++)
scanf("%d",&G[i][j]);
    //visited is initialized to zero
   for(i=0;i<n;i++)
        visited[i]=0;
    printf("The Depth First Search traversal is given be
    DFS(0);
}
void DFS(int i)
{
```

```
Enter number of vertices:4

Enter adjecency matrix of the graph:
0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0
The Depth First Search traversal is given below.
0 1 2 3
--------------------------------
Process exited after 30.1 seconds with return value 0
Press any key to continue . . .
```

```
Compilation results...
--------
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\manas\OneDrive\Desktop\DFS.exe
- Output Size: 129.3408203125 KiB
- Compilation Time: 0.30s
```