

## TASK # 9

### Q1:

```
#include<iostream>
```

```
using namespace std;
```

```
//It is code for printing tree to check my other functions are  
working correctly or not
```

```
struct Trunk
```

```
{
```

```
Trunk *prev;
```

```
string str;
```

```
Trunk(Trunk *prev, string str)
```

```
{
```

```
this->prev = prev;
```

```
this->str = str;
```

```
}
```

```
};
```

```
// Helper function to print branches of the binary tree
```

```
void showTrunks(Trunk *p)
```

```
{
```

```
if (p == NULL)
```

```
return;
```

```
showTrunks(p->prev);
```

```
std::cout<< p->str;
```

```
}
```

```
class Employ_node
```

```
{
```

```
public:
```

```
int emp_id;
```

```
string emp_name;
```

```
string emp_department;
```

```
string emp_designation;  
Employ_node *left;  
Employ_node *right;
```

```
Employ_node(int id, string name, string department, string  
designation)  
{  
    emp_id = id;  
    emp_name = name;  
    emp_department = department;  
    emp_designation = designation;  
    left = NULL;  
    right = NULL;  
}  
};
```

```
class Employ_Tree  
{  
public:  
    Employ_node *root;  
    Employ_Tree()  
    {  
        root = NULL;  
    }  
}
```

```
void Empoly_insertion(Employ_node *root2, int id, string name,  
string department, string designation)  
{  
    if(root == NULL)  
    {  
        root = new Employ_node(id, name, department, designation);  
        return;  
    }
```

```
    if(id < root2->emp_id)  
    {  
        if(root2->left == NULL)  
        {  
            root2->left = new Employ_node(id, name, department,  
designation);  
        }
```

```

else
{
Empoly_insertion(root2->left, id, name, department,
designation);
}
}
else
{
if(root2->right == NULL)
{
root2->right = new Employ_node(id, name, department,
designation);
}
else
{
Empoly_insertion(root2->right, id, name, department,
designation);
}
}
return;;
}

```

```

void Empoly_searching(Employ_node *passed_root, int
Empoly_id)
{
if(root == NULL)
{
cout<<"No Empolyee is present in the company"<<endl;
return;
}
if(Empoly_id == passed_root->emp_id)
{
cout<<"Empolyee ID: "<<passed_root->emp_id<<endl;
cout<<"Empolyee Name: "<<passed_root-
>emp_name<<endl;
cout<<"Empolyee Department: "<<passed_root-
>emp_department<<endl;
cout<<"Empolyee Designation: "<<passed_root-
>emp_designation<<endl;
return;
}
}

```

```

else if(Empoly_id < passed_root->emp_id)
{
if(root->left == NULL)
{
cout<<"Empolyee with ID "<<Empoly_id<<" is not present in
the company"<<endl;
return;
}
else
{
Empoly_searching(passed_root->left, Empoly_id);
}
}
else
{
if(passed_root->right == NULL)
{
cout<<"Empolyee with ID "<<Empoly_id<<" is not present in
the company"<<endl;
return;
}
else
{
Empoly_searching(passed_root->right,Empoly_id);
}
}
}

```

```

//Empoly with same level of authority
int getLevel(Employ_node *root, int value, int level)
{
if(root==NULL)
{
return 0;
}

if(root->emp_id==value)
{
return level;
}

```

```

int l=getLevel(root->left,value,level+1);
if(l!=0)
{
return l;

}
else
{
int r=getLevel(root->right,value,level+1);
return r;
}

}
void Printing_particular_level(Employ_node *root2, int level)
{
if(root == NULL)
{
cout<<"No Employee is present in the company"<<endl;
return;
}
if(level == 0)
{
cout<<"Employee ID: "<<root2->emp_id<<endl;
cout<<"Employee Name: "<<root2->emp_name<<endl;
cout<<"Employee Department: "<<root2->emp_department<<endl;
cout<<"Employee Designation: "<<root2->emp_designation<<endl;
return;
}
else
{
if(root2->left != NULL)
{
Printing_particular_level(root2->left, level-1);
}
if(root2->right != NULL)
{
Printing_particular_level(root2->right, level-1);
}
}
}

```

```
}
```

```
//Total Depth of Employee Hierarchy
int Total_Depth_of_the_Employee_Hierarchy(Employ_node* n)
{
    if (n == NULL)
        return -1;
    else
    {
        int left_depth = Total_Depth_of_the_Employee_Hierarchy(n->left);
        int right_depth = Total_Depth_of_the_Employee_Hierarchy(n->right);
        if (left_depth > right_depth)
            return(left_depth + 1);

        else
            return(right_depth + 1);
    }
}
```

```
Employ_node* Value_and_Sub_tree(Employ_node *n, int value)
{
    if(n==NULL)
    {
        return NULL;
    }
    if(n->emp_id==value)
    {
        return n;
    }
    Employ_node *curr;
    curr=Value_and_Sub_tree(n->left,value);
    if(curr!=NULL)
        return curr;
    curr=Value_and_Sub_tree(n->right,value);
    if(curr!=NULL)
        return curr;
```

```
return NULL;
}
```

```
void Display_All_empoly_details(Employ_node *passed_root)
{
if(root==NULL)
{
cout<<"Company has no empoly information"<<endl;
return;
}
if(passed_root==NULL)
return;
```

```
if(passed_root->right ==NULL)
{
cout<<"*****\n";
cout<<"Empoly ID: "<<passed_root->emp_id<<endl;
cout<<"Empoly Name: "<<passed_root->emp_name<<endl;
cout<<"Empoly Department: "<<passed_root->emp_department<<endl;
cout<<"Empoly Designation: "<<passed_root->emp_designation<<endl;
cout<<"*****\n";
```

```
Display_All_empoly_details(passed_root->left);
return;
}
else
Display_All_empoly_details(passed_root->right);
```

```
cout<<"Empoly ID: "<<passed_root->emp_id<<endl;
cout<<"Empoly Name: "<<passed_root->emp_name<<endl;
cout<<"Empoly Department: "<<passed_root->emp_department<<endl;
cout<<"Empoly Designation: "<<passed_root->emp_designation<<endl;
```

```
if(passed_root->left!=NULL)
{
```

```
Display_All_empoly_details(passed_root->left);  
}
```

```
return;  
}  
};
```

```
void printTree(Employ_node *root, Trunk *prev, bool isRight)  
{  
if (root == NULL)  
return;  
string prev_str = " ";  
Trunk *trunk = new Trunk(prev, prev_str);
```

```
printTree(root->right, trunk, true);
```

```
if (!prev)  
trunk->str = "---";  
else if (isRight)  
{  
trunk->str = ".---";  
prev_str = " |";  
}  
else  
{  
trunk->str = "`---";  
prev->str = prev_str;  
}
```

```
showTrunks(trunk);  
cout<< root->emp_id << endl;
```

```
if (prev)  
prev->str = prev_str;  
trunk->str = " |";
```

```
printTree(root->left, trunk, false);  
}  
int main(void)  
{
```



```

//As we are using BST so the empoly ID magnitude will decide
which employ is senior and which is junior
Employ_Tree DATA_empoly;
while(1)
{
cout<<"*****Managing Employee
Hierarchy*****\n";
cout<<"1)Add Empoly\n2)Search for Employ by ID\n3)Finding
Employees with Same Authority\n4)Total Depth of the
Employee Hierarchy\n5)Finding Subordinates of an Employee\n
n6)Viewing All Employee Details\n7)End\nUser_Input:";
int user_input;
int emp_id;
string emp_dept;
string emp_designation;
string emp_name;
cin>>user_input;
switch (user_input)
{
case 1:
cout<<"*****\n";
cout<<"Enter Employee ID: ";
cin>>emp_id;
cout<<"Enter Employee Name: ";
cin>>emp_name;
cout<<"Enter Employee Department:";
cin>>emp_dept;
cout<<"Enter Employee Designation:";
cin>>emp_designation;
DATA_empoly.Empoly_insertion(DATA_empoly.root,emp_id,
emp_name, emp_dept,emp_designation);
cout<<"*****\n";
break;
case 2:
cout<<"*****\n";
cout<<"Enter Employee ID:";
cin>>emp_id;
DATA_empoly.Empoly_searching(DATA_empoly.root,emp_id);

```

```

cout<<"*****\n";
break;
case 3:
cout<<"*****\n";
cout<<"Enter the ID of Employ to find Employees With same
Authority:";
cin>>emp_id;
DATA_empoly.Printing_particular_level(DATA_empoly.root,DATA_
empoly.getLevel(DATA_empoly.root,emp_id,0));
cout<<"*****\n";
break;
case 4:
cout<<"*****\n";

cout<<"Total Depth of the employee
Hierarchy:"<<DATA_empoly.Total_Depth_of_the_Employee_Hier
archy(DATA_empoly.root)+1;
cout<<"*****\n";

break;
case 5:
cout<<"*****\n";
cout<<"Enter ID to find Subordinate Empolyee:";
cin>>emp_id;
cout<<"Employee and his/her Subordinates:"<<endl;
DATA_empoly.Display_All_empoly_details(DATA_empoly.Value_a
nd_Sub_tree(DATA_empoly.root,emp_id));
printTree(DATA_empoly.Value_and_Sub_tree(DATA_empoly.root,e
mp_id),NULL,false);
cout<<"*****\n";

break;

case 6:

```

```

cout<<"*****\n";
7
DATA_empoly.Display_All_empoly_details(DATA_empoly.root);
printTree(DATA_empoly.root,NULL,false);
break;
case 7:
return 0;
default:
break;
}
}
return 0;
}

```

Output:

(1)

```

*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:

```

(2)

```
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:1
*****
Enter Employee ID: 10
Enter Employee Name: Raees
Enter Employee Department:CS
Enter Employee Designation:CEO
*****
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:1
*****
Enter Employee ID: 5
Enter Employee Name: Khan
Enter Employee Department:cs
Enter Employee Designation:HR
*****
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:1
*****
Enter Employee ID: 15
Enter Employee Name: balla
Enter Employee Department:AI
Enter Employee Designation:AI
*****
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User Input:█
```

3)

```
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:2
*****
Enter Employee ID:15
Empolyee ID: 15
Empolyee Name: balla
Empolyee Department: AI
Empolyee Designation: AI
*****
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:2
*****
Enter Employee ID:10
Empolyee ID: 10
Empolyee Name: Raees
Empolyee Department: CS
Empolyee Designation: CEO
*****
```

4)

```
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:3
*****
Enter the ID of Employ to find Employees With same Authority:1
Empolyee ID: 1
Empolyee Name: ASAF
Empolyee Department: CS
Empolyee Designation: CEO
Empolyee ID: 20
Empolyee Name: Bilal
Empolyee Department: CS
Empolyee Designation: HR
*****
```

5)

```
*****Managing Employee Hierarchy*****
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:4
*****
Total Depth of the employee Hierarchy:5*****
```

6)

```
*****Managing Employee Hierarchy**
1)Add Empoly
2)Search for Employ by ID
3)Finding Employees with Same Authority
4)Total Depth of the Employee Hierarchy
5)Finding Subordinates of an Employee
6)Viewing All Employee Details
7)End
User_Input:5
*****
Enter ID to find Subordinate Employee:5
Employee and his/her Subordinates:
*****
Empoly ID: 5
Empoly Name: Khan
Empoly Department: cs
Empoly Designation: HR
*****
*****
Empoly ID: 4
Empoly Name: Abbas
Empoly Department: asd
Empoly Designation: qwe
*****
*****
Empoly ID: 2
Empoly Name: tahir
Empoly Department: cs
Empoly Designation: CEO
Empoly ID: 1
Empoly Name: ASAF
Empoly Department: CS
Empoly Designation: CEO
*****
*****
Empoly ID: 0
Empoly Name: naeem
Empoly Department: cs
Empoly Designation: as
*****
---5
|      .---4
|      .---2
|      \---1
|      \---0
*****
```

7)

```
*****
Empoly ID: 20
Empoly Name: Bilal
Empoly Department: CS
Empoly Designation: HR
*****
Empoly ID: 15
Empoly Name: balla
Empoly Department: AI
Empoly Designation: AI
Empoly ID: 10
Empoly Name: Raees
Empoly Department: CS
Empoly Designation: CEO
*****
Empoly ID: 5
Empoly Name: Khan
Empoly Department: cs
Empoly Designation: HR
*****
*****
Empoly ID: 4
Empoly Name: Abbas
Empoly Department: asd
Empoly Designation: qwe
*****
Empoly ID: 2
Empoly Name: tahir
Empoly Department: cs
Empoly Designation: CEO
Empoly ID: 1
Empoly Name: ASAF
Empoly Department: CS
Empoly Designation: CEO
*****
Empoly ID: 0
Empoly Name: naeem
Empoly Department: cs
Empoly Designation: as
*****
      .---20
    .---15
  ---10
    \---5
      |   .---4
      |   .---2
      |   \---1
```