```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct node {

    int id;

    char name[20];

    int time;

    struct node* right;

    struct node* left;

};


typedef struct node treenode;


treenode* root = NULL;


treenode* newnode(int id, char name[], int time) {

    treenode* temp = (treenode*)malloc(sizeof(treenode));

    temp->id = id;

    strcpy(temp->name, name);

    temp->time = time;

    temp->right = NULL;

    temp->left = NULL;

    return temp;

}


treenode* createbst(treenode* root, int id, char name[], int time) {

    if (root == NULL) {

        return newnode(id, name, time);
    } else if (id < root->id) {
        root->left = createbst(root->left, id, name, time);
    } else if (id > root->id) {
        root->right = createbst(root->right, id, name, time);
    } else {
        printf("Duplicate id not allowed\n");
    }
    return root;
}

void inorder(treenode* root) {
    if (root == NULL)
        return;
    inorder(root->left);
    printf("%d\t\t%s\t\t\t%d \n", root->id, root->name, root->time);
    inorder(root->right);
}

int main() {
    int i, n, id, time;
    char name[20];
    printf("Enter the number of employees: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Enter employee id: ");
        scanf("%d", &id);
        printf("Enter employee name: ");
        scanf("%s", name);
        printf("Enter employee login time: ");
        scanf("%d", &time);
        root = createbst(root, id, name, time);
    }
    printf("\nID\t\tName\t\tLogin Time\n");
    inorder(root);
    return 0;
}
```

Output:

```

Enter the number of employees: 2
Enter employee id: 6
Enter employee name: fgh
Enter employee login time: 7
Enter employee id: 5
Enter employee name: fgh
Enter employee login time: 7

ID              Name            Login Time
5               fgh                    7
6               fgh                    7
```