

```

#include<stdio.h>

#include<stdlib.h>

#include<string.h>


struct node{
int id;
char name[20];
int time;
struct node *right,*left;
};

typedef struct node treenode;

treenode *root = null;


treenode* newnode(int id,char name[],int time){
treenode* temp = (treenode*)malloc(sizeof(treenode));

int i;

temp->id = id;
for( i=0;i<strlen(name);i++)
temp->name[i] = name[i];
temp->name[i] = '\0';
temp->time = time;
temp->right = null;
temp->left = null;
return temp;
}


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

```

```

if(root==null){
treenode* p = newnode(id,name,time);
root = p;
return root;
}
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%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\n");
scanf("%d",&n);

```

```
for(i=0;i<n;i++){  
    printf("enter employee id\n");  
    scanf("%d",&id);  
  
    printf("enter employee name\n");  
    scanf("%s",name);  
  
    printf("enter employee login time\n");  
    scanf("%d",&time);  
  
    root = createbst(root,id,name,time);  
}  
printf("id\t\t name\t\t\t login time\n");  
inorder(root);  
}
```