Question 1:

#include<stdio.h>

#include<stdlib.h>

#include <string.h>

typedef struct node

{

char srn[13];

struct node\*llink;

struct node\*rlink;

}node\_t;

typedef struct tree

{

node\_t\* root;

}tree\_t;

void init(tree\_t\*pt)

{

pt->root=NULL;

}

void create(tree\_t\*pt,int n)

{

node\_t\*temp;

node\_t\*pres;

node\_t\*prev;

printf("Enter root node: ");

pt->root=(node\_t\*)malloc(sizeof(node\_t));

scanf("%s",(pt->root->srn));

pt->root->llink=pt->root->rlink=NULL;

do

{

printf("Enter node value: ");

temp=(node\_t\*)malloc(sizeof(node\_t));

scanf("%s",(temp->srn));

temp->llink=temp->rlink=NULL;

prev=NULL;

pres=pt->root;

while(pres!=NULL)

{

prev=pres;

if(strcmp(temp->srn,pres->srn)<0)

pres=pres->llink;

else

pres=pres->rlink;

}

if(strcmp(temp->srn,prev->srn)<0)

prev->llink=temp;

else

prev->rlink=temp;

--n;

} while(n>0);

}

void inorder\_traversal(node\_t\*p)

{

if(p!=NULL)

{

inorder\_traversal(p->llink);

printf("%s\n",p->srn);

inorder\_traversal(p->rlink);

}

}

int search(node\_t\*p,char\* ele)

{

int found=0;

if(p!=NULL)

{

search(p->llink,ele);

if(strcmp(p->srn,ele)==0)

{

found=1;

return(found);

}

search(p->rlink,ele);

}

return(found);

}

int main()

{

tree\_t t;

init(&t);

printf("Enter number of nodes: ");

int len;

scanf("%d",&len);

create(&t,len-1);

inorder\_traversal(t.root);

printf("\n Enter srn to search for: ");

char ele[13];

scanf("%s",ele);

int res=search(t.root,ele);

if(res==1)

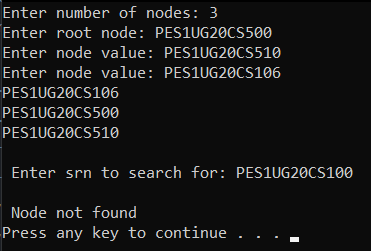
printf("\n %s is found \t",ele);

else

printf("\n Node not found");

return 0;

}



Question 2:

#include<stdio.h>

typedef struct tree\_node

{

int info;

int used;

}TREE;

#define MAXNODES 50

void init(TREE t[MAXNODES])

{

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

t[i].used=0;

}

int create(TREE \*bst)

{

int ele, wish;

printf("Enter the root element\n");

scanf("%d",&bst[0].info);

bst[0].used=1;

int cnt=1;

do{

printf("Enter an element\n");

scanf("%d",&ele);

int p=0;

while(p<MAXNODES && bst[p].used)

{

if(ele<bst[p].info)

p=2\*p+1;

else

p=2\*p+2;

}

if(p>=MAXNODES)

printf("Insertion not possible\n");

else

{

bst[p].info=ele;

bst[p].used=1;

cnt++;

}

printf("Do you wish to add another\n");

scanf("%d",&wish);

}while(wish);

return cnt/2;

}

void preorder(TREE\* bst, int r)

{

if(bst[r].used)

{

printf("%d ",bst[r].info);

preorder(bst,2\*r+1);

preorder(bst,2\*r+2);

}

}

int main()

{

TREE bst[MAXNODES];

init(bst);

int height=create(bst);

printf("The height of the tree is %d and the level is %d\n",height+1,height);

preorder(bst,0);

return 0;

}

