

Napisati program koji učitava cele brojeve sve dok ne učitava 0. Od učitanih brojeva formirati uređeno binarno stablo. Iz formiranog stabla ispisati brojeve u neopadajućem poretku.

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

struct drvo{
    int broj;
    struct drvo *levi,*desni;};

#define novi(x) x=(struct drvo *) malloc(sizeof(struct drvo))

void dodaj(struct drvo **,int);
struct drvo* form();
void ispis(struct drvo*);

main(){
    struct drvo *p;
    int n,m;
    p=form();
    ispis(p);}
```

```

void dodaj(struct drvo **p,int k){
    struct drvo *temp,*pom1,*pom2;
    novi(temp);
    if(!temp) {
        printf("\nGreska pri alokaciji memorije\n");
        exit(0);
    }
    temp->broj=k;
    temp->levi=temp->desni=NULL;
    if (!(*p)) *p=temp;
    else{
        pom1=*p;
        while(pom1){
            pom2=pom1;
            if(k < pom1->broj) pom1=pom1->levi;
            else pom1=pom1->desni;
        }
        if(k < pom2->broj) pom2->levi=temp;
        else pom2->desni=temp;
    }
}

```

```
struct drvo* form(){
    struct drvo *koren;
    int k;
    koren=NULL;
    scanf("%d",&k);
    while(k) {
        dodaj(&koren,k);
        scanf("%d",&k);
    }
    return koren;
}

void ispis(struct drvo *p){
    if (p) {
        ispis(p->levi);
        printf("%5d",p->broj);
        ispis(p->desni);
    }
}
```

a) Odrediti maksimum učitanih brojeva

```
int maksimum(struct drvo *p){  
    while (p)  
    {  
        if (p->desni) p=p->desni;  
        else  
            return(p->broj);  
    }  
}
```

b) Izračunati zbir elemenata u formiranom binarnom stablu

```
int zbir(struct drvo *p){  
    int s;  
    s=p->broj;  
    if(p->levi) s+=zbir(p->levi);  
    if(p->desni) s+=zbir(p->desni);  
    return s;  
}
```

c) Za formirano binarno stablo odrediti dubinu (maksimalan nivo)

```
int dubina(struct drvo *p){
    int dl=0,dd=0;
    if(p){
        if (p->levi) dl=dubina(p->levi);
        if (p->desni) dd=dubina(p->desni);
        if (dl>dd) return ++dl;
        else return ++dd;
    }
    else return 0;
}
```

- d) Za uneti broj X ispitati da li se nalazi u formiranom binarnom stablu i ako se nalazi, na kom nivou se prvi put pojavljuje.

```
int nadj(i(struct drvo *p,int k){
    int n=0;
    if(p) {
        if (k==p->broj) return 1;
        else if (k<p->broj) n=nadj(i(p->levi,k);
        else n=nadj(i(p->desni,k);
    }
    if (n) return ++n;
    else return 0;
}
```