

Binarna stabla

SPA2

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 poretaku.

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

Napisati program za formiranje binarnog stabla koje sadrži izraz sa 4 osnovne operacije(+, -, *, /), pri čemu su prisutne sve zagrade, tj. (a op b)

Vrednost izraza

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

struct drvo{
    union {
        int broj;
        char oper;
    } sadrzaj;
    struct drvo *levi,*desni;};

struct stek {
    struct drvo *sadrzaj;
    struct stek *rep;};

#define novis(x) x=(struct stek *)malloc(sizeof(struct stek))
#define novid(x) x=(struct drvo *)malloc(sizeof(struct drvo))
```

Vrednost izraza

```
struct stek* push(struct stek *p, struct drvo *d){
    struct stek *temp;
    novis(temp);
    if(!temp) exit(0);
    temp->sadrzaj=d;    temp->rep=p;
    p=temp;
    return p; }
```

```
struct drvo* pop(struct stek **p){
    struct stek *temp;
    struct drvo *d;
    temp=*p;    *p=(*p)->rep;
    d=temp->sadrzaj;
    free(temp);
    return d;}
```

```
int isoper(char c){
    if((c=='+') || (c=='-') || (c=='*') || (c=='/')) return 1;
    return 0;}
```

Vrednost izraza

```
struct drvo* napravi(){
    char s[50],ch,op;
    int br=0;
    struct drvo *temp,*l,*d,*o;
    struct stek *poc=NULL;
    while((ch=getchar())!='\n'){
        if (isdigit(ch)){
            while(isdigit(ch)) {
                s[br++]=ch;
                ch=getchar();
            }
            s[br]='\0';
            br=0;
            novid(temp);
            if(!temp) exit(0);
            temp->sadrzaj.broj=atoi(s);
            temp->levi=temp->desni=NULL;
            poc=push(poc,temp);
        }
    }
}
```

...

Vrednost izraza

...

```
    if(isoper(ch)) {
        novid(temp);
        if(!temp) exit(0);
        temp->sadrzaj.oper=ch;
        temp->levi=temp->desni=NULL;
        poc=push(poc,temp);
    }
    if(ch=='('){
        d=pop(&poc);
        o=pop(&poc);
        l=pop(&poc);
        o->levi=l;
        o->desni=d;
        poc=push(poc,o);
    }
}
return poc->sadrzaj;
```

```
}
```

Vrednost izraza

```
int vrednost(struct drvo *p){
    int r,d;
    if((p->levi) && (p->desni)){
        switch(p->sadrzaj.oper){
            case '+' : r=vrednost(p->levi)+vrednost(p->desni); break;
            case '-' : r=vrednost(p->levi)-vrednost(p->desni); break;
            case '*' : r=vrednost(p->levi)*vrednost(p->desni); break;
            case '/' : d=vrednost(p->desni);
                       if (d) r=vrednost(p->levi)/d;
                       else exit(0);
                       break;
        }
    }
    else r=p->sadrzaj.broj;
    return r;
}
```

Vrednost izraza

```
void ispis(struct drvo *p){
    if ((p->levi) && (p->desni)){
        printf("(");
        ispis(p->levi);
        printf(" %c ",p->sadrzaj.oper);
        ispis(p->desni);
        printf(")");
    }
    else printf("%d",p->sadrzaj.broj);
}

main(){
    struct drvo *exp;
    exp=napravi();
    ispis(exp);
    printf("\nVrednost je %d\n",vrednost(exp));
}
```

Domaći

- Napisati program koji iz ulazne datoteke učitava podatke o studentima
 - Ime
 - Prezime
 - Godina upisa
 - Godina studija
 - Prosek

Ulazne podatke organizovati u obliku binarnog stabla sortiranog po prezimenu studenta. Za uneto prezime ispisati podatke svih studenata sa tim prezimenom

Domaći

- Koristeći kontrole dozvoliti da budu unete promenljive ali ih pri izračunavanju vrednosti izraza ne konstatovati.
(ovaj deo ćete dobiti)
- Popraviti funkciju VREDNOST tako da prihvata vrednosti promenljivih pri izračunavanju vrednosti izraza