

Povezane liste

SPA2

Napisati funkciju koja iz povezane liste brojeva izbacuje elemente veće od broja k

```
void izbaci(struct lista **p, int k){
    struct lista *res,*pos,*temp;
    res=*p;  pos=res;
    while(pos->broj>k) {
        res=pos->rep;
        free(pos);
        pos=res;
    }
    while(pos->rep)
        if (pos->rep->broj>k) {
            temp=pos->rep;
            pos->rep=temp->rep;
            free(temp);
        }
        else pos=pos->rep;
    *p=res; }
```

Napisati funkciju koja u povezanu listu brojeva ubacuje element jednak broju k , iza svakog elementa koji je jednak broju k . Ukoliko je povezana lista prazna element k ubaciti kao jedini element liste.

```
void ubaci(struct lista **p, int k){
    struct lista *pos,*temp,*res;
    pos=*p;
    if (pos==NULL)
        {
            novi(pos);
            if (pos==NULL)
                {
                    printf("Greska\n");
                    exit(0);
                }
            pos->broj=k;
            pos->rep=NULL;
            *p=pos;
        }
}
```

```
else {
    while(pos)
        if (pos->broj==k)
            {
                novi(temp);
                if ( temp == NULL )
                    {
                        printf( "Greska pri alociranju memorije. \n");
                        exit(0);
                    }
                temp->broj=k;
                temp->rep=pos->rep;
                pos->rep=temp ;
                pos=temp->rep;
            }
        else pos=pos->rep;
    }
}
```

- Definirati strukturu za rad koja predstavlja povezanu listu koja sadrži sledeće podatke o studentima:
 - Ime
 - Prezime
 - Broj indeksa
 - Godina studija
 - Prosek

Napisati program koji za uneti prirodan broj n i podatke za n studenata formira:

sortiranu listu, ali tako da se na početku liste nalaze studenti sa najvećim prosekom, pri čemu za studente koji imaju isti prosek prednost imaju oni koji su na višoj godini studija. Na izlazu ispisati podatke za one studente koji imaju prosek veći od zadatog.

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

```
struct studenti{
    char *ime,*prezime,*br_ind;
    int godina;
    float prosek;
};
```

```
struct lista{
    struct studenti *glava;
    struct lista *rep;
};
```

```
#define novi(x) x=(struct lista *)malloc(sizeof(struct lista))
```

```
void ispis(struct lista *p){
struct lista *pom;
pom=p;
struct studenti *st;
    while(pom) {
        st=pom->glava;
        printf("%s  %s  ",st->br_ind,st->prezime);
        printf("%s  ",st->ime);
        printf("%d  %f\n",st->godina,st->prosek);
        pom=pom->rep;
    }
}
```



```
void dodaj_sort(struct lista **p,struct studenti *q){
    struct lista *temp,*res,*pos;
    novi(temp);
    if (!temp){
        printf("Greska pri alokaciji memorije\n");
        exit(0);
    }
    temp->glava=q;
    temp->rep=NULL;
    res=*p;
    if(!res) res=temp;
    ...
}
```

```

...
else{
    if ((res->glava->prosek<q->prosek) ||
        ((res->glava->prosek==q->prosek) &&
         (res->glava->godina<q->godina))){
        temp->rep=res;
        res=temp;
    }
    else {
        pos=res;
        while((pos->rep) &&
              ((pos->rep->glava->prosek>q->prosek) ||
               ((pos->rep->glava->prosek==q->prosek) &&
                (pos->rep->glava->godina>q->godina))))
            pos=pos->rep;
        temp->rep=pos->rep;
        pos->rep=temp;
    }
}
*p=res;
}

```

```
struct studenti* ucitaj_studenta(){
    struct studenti *st;
    char s[100];
    getchar();
    st=(struct studenti*)malloc(sizeof(struct studenti));
    printf("indeks: "); scanf("%s",s);
    st->br_ind=(char*)malloc(strlen(s)+1);
    strcpy(st->br_ind,s);
    printf("prezime: "); scanf("%s",s);
    st->prezime=(char*)malloc(strlen(s)+1);
    strcpy(st->prezime,s);
    printf("ime: "); scanf("%s",s);
    st->ime=(char*)malloc(strlen(s)+1);
    strcpy(st->ime,s);
    printf("godina studija: "); scanf("%d",&st->godina);
    printf("prosek: "); scanf("%f",&st->prosek);
    return st;
}
```

```
struct lista *form_sort(){
    int i,n;
    struct lista *p;
    struct studenti *q;
    p=NULL;
    scanf("%d",&n);
    for(i=0;i<n;i++){
        q=ucitaj_studenta();
        printf("Ucitan\n");
        dodaj_sort(&p,q);
    }
    return p;
}
```

```
main(){
    struct lista *p;
    p=form_sort();
    ispis(p);
}
```