

Ime i prezime	Broj indeksa	Broj poena

1. (1 poen) Data je struktura:

```
struct trougao {  
    float a,b,c;  
};
```

Definisati funkciju **struct trougao napravi(float a, float b, float c)** koja formira strukturu trougla, dodeljuje joj vrednosti koje prihvata preko argumenata i vraća dobijenu strukturu.

```
struct trougao napravi(float a, float b, float c) {  
    struct trougao t;  
    t.a = a;  
    t.b = b;  
    t.c = c;  
    return t;  
}
```

2. (0.5 poena) Promenljivu **struct trougao X**, za strukturu datu u zadatku 1, inijalizovati vrednostima 1, 2 i 2.5 koristeći funkciju **napravi** iz zadatka 1.

```
struct trougao X = napravi(1, 2, 2.5);
```

3. (1.5 poena) Data je struktura

```
struct niske {  
    int poz;  
    char *str; };
```

i promenljiva

```
struct niske p[]={0, "Strukture", 1, "podataka", 0, "jedan"};
```

Šta će biti vrednosti sledećih konstrukcija:

- a) `p[0].str`                      **Strukture**  
b) `p[1].str[++p[1].poz]`        **d**  
c) `*(++p[2].str)`                **e**

4. (1 poen) Data je struktura

```
struct djak {  
    char ime[10];  
    char oznaka;  
    union {  
        char opisna;  
        int vrednost;  
    } OCENA;  
} DJAK[2];
```

Elementu `DJAK[0]` dodeliti opisnu ocenu A, a elementu `DJAK[1]` numeričku ocenu 5.

```
DJAK[0].OCENA.opisna = 'A';  
DJAK[1].OCENA.vrednost = 5;
```

5. (1 poen) Sortiraj niz brojeva 17 3 22 13 78 28 30 54 47 koristeći **Merge sort** i ispisati svaki korak prilikom sortiranja brojeva.

```

17 3 22 13 78 28 30 54 47
3 17 22 13 78 28 30 54 47
3 13 17 22 78 28 30 54 47
3 13 17 22 28 30 54 78 47
3 13 17 22 28 30 47 54 78

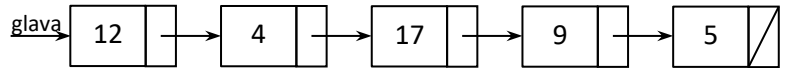
```

6. (1.5 poena) Data je struktura:

```

struct element {
    int vrednost;
    struct element* sledeci;
};
struct element* glava;

```



i primer liste na slici.

Napisati niz komandi kojima se u listu sa slike dodaje element sa vrednošću 10, između prvog i drugog elementa liste.

```

struct element *novi = (struct element *)malloc(sizeof(struct element));
novi->vrednost = 10;
novi->sledeci = glava->sledeci;
glava->sledeci = novi;

```

7. (1.5 poena) Za strukturu iz zadatka 6 napisati funkciju koja dodaje element ispred (levo od) element koji je prvi veći od vrednosti koja se dodaje.

Primer: 1 9 5 14 3, dodaje se broj 8

Izlaz: 1 8 9 5 7 14 3

```

struct element *dodaj(struct element *glava, int vrednost) {
    struct element *pom = glava, *novi;

    novi = (struct element *)malloc(sizeof(struct element));
    novi->vrednost = vrednost;
    novi->sledeci = NULL;

    if (pom == NULL)
        glava = novi;
    else {
        while (pom->sledeci && vrednost >= pom->sledeci->vrednost)
            pom = pom->sledeci;

        novi->sledeci = pom->sledeci;
        pom->sledeci = novi;
    }

    return glava;
}

```

8. (2 poena) Za strukturu iz zadatka 6 napisati funkciju koja za dati pokazivač na početak liste briše prvi element koji je **manji** od **srednje vrednosti** svih elemenata.

```
struct element *obrisi(struct element *glava) {
    int br = 0;
    float avg = 0.0;
    struct element *pom = glava, *pret;

    while (pom) {
        avg += pom->vrednost;
        br++;
        pom = pom->sledeci;
    }
    avg /= br;

    pret = NULL;
    pom = glava;
    while (pom && pom->vrednost >= avg)
        pom = pom->sledeci;

    if (pret == NULL)
        glava = glava->sledeci;
    else
        pret->sledeci = pom->sledeci;

    free(pom);
    return glava;
}
```