

Strukture podataka i algoritmi 1
II test – max 13 poena

21.05.2022.

Ime i prezime	Broj indeksa	Broj poena
---------------	--------------	------------

1. (1.5 poena) Dopuni sledeći kod tako da se u okviru printf-a ispišu potrebni podaci.

```
#include <stdio.h>
struct knjige{
    int broj_strana;
    float cena;
    char naziv[30];
    union {
        int godine;
        char ime[30];
        char prezime[30];
    }o_autoru;
};

void ucitaj(struct knjige *knjiga){
    scanf("%s %s %f", (knjiga)->naziv, (knjiga)->o_autoru.ime, &(knjiga)->cena);
}

int main(){
    struct knjige knjiga;
    ucitaj(&knjiga);
    printf("Nazivknjige:%s\nIme:%s\nCena:%f\n", knjiga.naziv,      knjiga.o_autoru.ime,
knjiga.cena);
    return 1;
}
```

2. (1.5 poen) Napisati program koji treba da pronađe sumu ili proizvod argumenata komandne linije, kojih može biti proizvoljno mnogo. Prilikom pokretanja se najpre, kao prvi argument, navodi "suma" ili "proizvod", a iza toga slede brojevi koje treba sumirati/pomnožiti.

Primer: ./zad suma 1 2 3 4

Izlaz: 10

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

int main(int argc, char **argv) {

    if (argc == 1) {
```

```

        printf("Greska!\n");
        return 1;
    }

    int n = argc - 2;
    int i;
    int rezultat;

    if (strcmp(argv[1], "suma") == 0) {
        rezultat = 0;

        for (i = 0; i < n; i++) {
            rezultat += atoi(argv[i + 2]);
        }
    }
    else if (strcmp(argv[1], "proizvod") == 0) {
        rezultat = 1;

        for (i = 0; i < n; i++) {
            rezultat *= atoi(argv[i + 2]);
        }

        if (n == 0) {
            rezultat = 0;
        }
    }
    else {
        printf("Pogresan argument!\n");
        return 1;
    }

    printf("%d\n", rezultat);

    return 0;
}

```

3. (1 poen) U navedeni kod dodati komande kojima se na izlazu ispisuju svi brojevi iz zadatog opsega. U **funkcija**-i nije dozvoljeno koristiti printf, već **funkcija_zdravo**. **Napomena:** Koristiti pokazivače na funkcije.

```

#include <stdio.h>
void funkcija_zdravo (int n) {
    printf("%d\n",n);
}

```

```

    }
    void funkcija(void (*f) (int), int x,int y)
    {
        int i;
        for(i=x;i<=y;i++)
            (*f)(i);
    }
    int main() {

        int l,k;
        l=4;
        k=13;
        funkcija(&funkcija_zdravo,l,k);
        return 0;
    }

```

4. (2 poena) Šta je rezultat sledećeg koda?

```

#include <stdio.h>
int main()
{
    int matrica[3][4] = {
        {10, 11, 12, 13},
        {20, 21, 22, 23},
        {30, 31, 32, 33}
    };

    int (*p)[4];
    p = matrica;
    printf("%d %d %d\n", **p, *(*p + 1) + 2), *(*p + 2) + 3));
    return 0;
}
10 22 33

```

5. (1 poen) Sortiraj niz brojeva 22 11 15 64 41 21 99 12 10 69 koristeći Merge sort algoritam i ispisati svaki korak prilikom sortiranja brojeva.

```

22 11 15 64 41 21 99 12 10 69
11 22 15 64 41 21 99 12 10 69
11 15 22 41 64 12 21 99 10 69
10 11 12 15 21 22 41 64 69 99

```

6. (1 poen) U sortiranom nizu iz prethodnog zadatka koristeći binarnu pretragu pronaći broj 41 i napisati korake pri pretrazi, odnosno granične vrednosti u oblastima pretrage u svakom koraku.

low=10 high=99 mid=21

low=22 high=99 mid=64

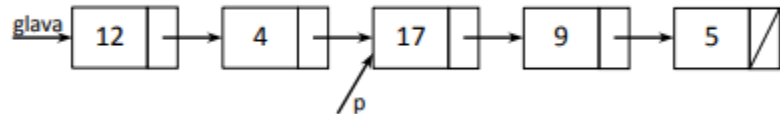
low=22 high=41 mid=41

7. (1 poen) Data je struktura:

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

```
struct element* glava;
```

Napisati niz komandi kojima će jedno mesto nakon elementa na koji pokazuje pokazivač p biti dodat element čija je vrednost 19, bez korišćenja petlji.



```
struct element *novi = (struct element *) malloc(sizeof(struct element));
```

```
struct element *pom = p->sledeci;
```

```
novi->vrednost = 19;
```

```
novi->sledeci=pom;
```

```
p->sledeci=novi;
```

8. (2 poena) Za strukturu iz zadatka 7 napisati funkciju koja dodaje element na mesto pre elementa u listi na koji pokazuje pokazivač p koji je prosleđen funkciji. Pored pokazivača p, funkciji je prosleđen i pokazivač na prvi element u listi.

```
void dodajPreP(struct element **glava, struct element *p, struct element *novi)
```

```
{  
    struct element *pom;  
    pom=*glava;  
    while(pom->sledeci!=p)  
    {  
        pom=pom->sledeci;  
    }  
    novi->sledeci=pom->sledeci;  
    pom->sledeci=novi;  
}
```

9. (2 poena) Za strukturu iz zadatka 7 napisati funkciju koja za dati pokazivač na početak liste vraća pokazivač na maksimalni parni element liste.

```
struct element *maxListe(struct element **p)
{
    if (*p == NULL) return NULL;
    else
    {
        struct element *pom = *p, *max;
        max=NULL;

        while(pom)
        {
            if(pom->vrednost%2==0)
            {
                if (max==NULL)
                    max=pom;
                else if(max->vrednost<pom->vrednost)
                    max=pom;
            }
            pom = pom->sledeci;
        }
        return max;
    }
}
```