

Strukture podataka i algoritmi 1
II test – max 13 poena

8.5.2021.

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

1. (1 poen) Data je struktura:

```
struct krug
{
    float poluprecnik;
    float obim;
    float povrsina;
};
```

Definisati funkciju **struct krug napravi(float r)** koja formira strukturu kruga, dodeljuje joj vrednosti koje prihvata preko argumenta i vraća dobijenu strukturu.

```
struct krug napravi(float r) {
    struct krug k;
    float pi = 3.14;

    k.poluprecnik = r;
    k.obim = 2 * r * pi;
    k.povrsina = r * r * pi;

    return k;
}
```

2. (0.5 poena) Odrediti zapreminu lopte čiji je poprečni presek dat promenljivom struct krug *x. Zapremina lopte računa se preko obrasca $V = \frac{4}{3}r^3\pi, \pi \approx 3.14$.

```
float zapremina = (4.0/3) * pow(x->poluprecnik, 3) * 3.14;
```

3. (1.5 poen) Napisati program koji kao argumente komandne linije prima dva broja **a** i **b** i jedan od karaktera **k** ili **s** i na osnovu toga računa vrednost, ukoliko je unet karakter **k** računa se **b**-ti koren iz **a**. U suprotnom, ako je unet karakter **s** računa se **b**-ti stepen broja **a**.

Primer: ./zad 16 4 k

Izlaz: 2

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

```

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

    if (argc != 4) {
        printf("Greska. Upotreba: %s prvi_broj drugi_broj stepen(s)/koren(k)\n", argv[0]);
        return 1;
    }

    int a, b;
    char op;
    int rezultat;

    a = atoi(argv[1]);
    b = atoi(argv[2]);
    op = *argv[3];

    if (op == 'k') {
        rezultat = round(pow(a, 1.0 / b));
    }
    else if (op == 's') {
        rezultat = pow(a, b);
    }
    else {
        printf("Moguće je samo stepenovanje ili korenovanje!\n");
        return 1;
    }

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

    return 0;
}

```

4. (1 poen) Šta je rezultat sledećeg koda?

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

```

union primer {
    int broj_int;
    int niz[30];
    float broj_float;
    struct {
        double a;
    } stepen;
    struct {
        double r;
        double P,O;
    } krug;
}

```

```

union {
    float niz[32];
    struct {
        int x,y,z;
    } trougao;
} primer2;
};

int main() {
    union primer unija;
    unija.broj_float = 12;
    unija.krug.r = 82;
    printf("%ld\n", sizeof(union primer));

    return 0;
}

```

128

5. (2 poena) Napisati program koji učitava matricu i određuje maksimalni element na sporednoj dijagonali. Memorijski prostor za matricu dinamički alocirati. Program implementirati bez korišćenja operatora [] prilikom unosa članova matrice i pronalaženja maksimalnog elementa.

```

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

```

```

int main() {

    int **matrica;
    int i, j;
    int n;
    int tren;

    scanf("%d", &n);

    matrica = (int **) malloc(n * sizeof(int *));
    for (i = 0; i < n; i++) {
        matrica[i] = (int *) malloc(n * sizeof(int));
    }

    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {

```

```

        scanf("%d", *(matrica + i) + j);
    }
}

int max =>(*matrica + (n - 1));
for (i = 1; i < n; i++) {
    tren =>(*matrica + i) + (n - i - 1));
    if (tren > max) {
        max = tren;
    }
}

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

return 0;
}

```

6. (1 poen) Sortiraj niz brojeva 39 69 99 48 83 36 33 70 23 77 koristeći Bubble sort i ispisati svaki korak prilikom sortiranja brojeva.

```

39 69 99 48 83 36 33 70 23 77
39 69 48 83 36 33 70 23 77 99
39 48 69 36 33 70 23 77 83 99
39 48 36 33 69 23 70 77 83 99
39 36 33 48 23 69 70 77 83 99
36 33 39 23 48 69 70 77 83 99
33 36 23 39 48 69 70 77 83 99
33 23 36 39 48 69 70 77 83 99
23 33 36 39 48 69 70 77 83 99
23 33 36 39 48 69 70 77 83 99

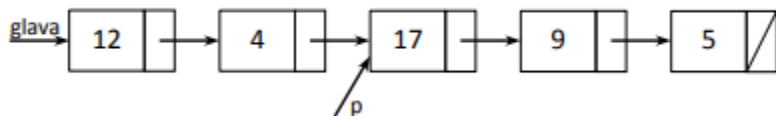
```

7. (1.5 poen) Data je struktura:

```

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

```



```

struct element* glava;

```

Napisati niz komandi kojima će dva mesta iza elementa (između elemenata 9 i 5) 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->sledeci;
pom->sledeci = novi;
```

8. (2 poena) Za strukturu iz zadatka 7 napisati funkciju koja dodaje element na pretposlednje mesto za ma koju listu, ako je funkciji prosleđen pokazivač na prvi element liste.

```
void dodajNaPretposlednjeMesto(struct element **p, int vrednost) {
    struct element* novi = (struct element *)malloc(sizeof(struct element));
    novi->vrednost = vrednost;
    novi->sledeci = NULL;
    struct element *pom = *p;
    struct element *preth = NULL;
    if (*p == NULL)
        *p = novi;
    else {
        while(pom->sledeci) {
            preth = pom;
            pom = pom->sledeci;
        }
        if (preth == NULL) {
            novi->sledeci = *p;
            *p = novi;
        }
        else {
            preth->sledeci = novi;
            novi->sledeci = pom;
        }
    }
}
```

9. (2.5 poena) Za strukturu iz zadatka 7 napisati funkciju koja za dati pokazivač na početak liste vraća pokazivač na element koji je jednak srednjoj vrednosti liste.

```
struct element *srednjaVrednostListe(struct element **p) {

    if (*p == NULL)
        return NULL;
    else {
        struct element *pom = *p;
        float srednjaVrednost = 0, suma = 0;
        int n = 0;
```

```
while(pom){
    n++;
    suma += pom->vrednost;
    pom = pom->sledeci;
}

srednjaVrednost = suma / n;

pom = *p;
while(pom && pom->vrednost != srednjaVrednost){
    pom = pom->sledeci;
}

return pom;
}
}
```