

Strukture podataka i algoritmi 1
Test – max 20 poena

Jun, 2020

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

1. (1.5) Šta je rezultat sledećih kodova?

```
int a = 27, b = 13, c = 18;  
printf("%d", (a & b) | c);
```

27

```
int a = 28;  
printf("%d", (a << 4) ^ 10);
```

458

```
int s = 0;  
for (b = 15; b; b >>= 1)  
    s += b;  
printf("%d", s);
```

26

2. (2.0) Sortiraj niz brojeva 17 3 22 13 78 28 30 54 47 6 koristeći Bubble sort i ispisati svaki korak prilikom sortiranja brojeva.

```
3 17 13 22 28 30 54 47 6 78  
3 13 17 22 28 30 47 6 54 78  
3 13 17 22 28 30 6 47 54 78  
3 13 17 22 28 6 30 47 54 78  
3 13 17 22 6 28 30 47 54 78  
3 13 17 6 22 28 30 47 54 78  
3 13 6 17 22 28 30 47 54 78  
3 6 13 17 22 28 30 47 54 78
```

3. (2.5) Data je struktura:

```
struct node{  
    float x;  
    struct node* next;  
};  
struct node* head;
```

Formirana je lista na čiji prvi element pokazuje promenljiva head i čiji elementi imaju vrednosti 1.2 2.3 4.5 3.2 10.0 7.8 12.1. Napisati niz komandi kojima se dobija lista 1.2 2.3 4.5 5.3 19.2 14.3 3.2 10.0 7.8 12.1.

```
struct node *temp, *novi;  
temp = head->next->next;  
for(i=0;i<3;i++){  
    novi=(struct node*)malloc(sizeof(struct node));  
    scanf("%f",&novi->x);  
    novi->next=temp->next;  
    temp->next=novi;  
    temp=novi;  
}
```

4. (2.5) Za strukturu iz zadatka 3 napisati funkciju koja dodaje element ispred (levo od) elementa koji je prvi jednak vrednosti koja se dodaje.

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

Izlaz: 1 9 5 5 7 14 3

```
struct node* dodaj(struct node *p, float val){
    struct node *temp, *novi;
    novi=(struct node*)malloc(sizeof(struct node));
    novi->x = val;
    if (p->x == val){
        novi->next = p;
        p = novi;
        return p;
    }
    temp = p;
    while(temp->next && temp->next->x != val)
        temp =temp->next;
    if(temp->next){
        novi->next=temp->next;
        temp->next=novi;
    }
}
```

5. (1.5) Napisati uslovni izraz koji odgovara sledećem kodu

```
if(x%4==0)
    y = x * 4;
else if(x%5==0)
    y = sqrt(x, 2);
else
    y = x * x;
```

$y = (x\%4)? (x\%5 ? x*x : \text{sqrt}(x)) : x*4;$

6. (1.0) Zaokružiti izraze/izraz koji su/je ekvivalatan/ni sa izrazom $a[i][j][k][l]$

a. $((*(a+i)+j)+k)+l$ b. $*(a+i)+j+k+l$
c. $*(((a+i)+j)+k+l)$ d. $*(a+i)+j+k+l$

7. (1.5) Koliko memorijskog prostora zauzima promenljiva koja je tipa struct geom_figura?

```
struct geom_figura {
    int tip;
    double koord[2];
    union {
        struct { double r; } kruznic;
        struct { double a; } kvadrat;
        struct { double a, b; } pravougaonik;
        struct { double a, b, c; } trougao;
    } figura;
};
```

$\text{sizeof}(\text{struct geom_figura}) \leq \text{sizeof}(\text{int}) + \text{sizeof}(\text{double}) + \text{sizeof}(\text{figura}) =$
 $= \text{sizeof}(\text{int}) + 2*\text{sizeof}(\text{double}) + 3*\text{sizeof}(\text{double})$

8. (2.5) Napisati program pomeri koji izvršava bitovsku operaciju pomeranja u levo za dva zadata broja x i y, tako što broj x pomeri u levo y puta. Brojevi x i y su predati prilikom pokretanja programa na sledeći način:
./pomeri 4 5

```
main(int argc, char **argv)
{
    int x,y;
    x = atoi(argv[1]);
    y = atoi(argv[2]);
    printf("%d %d %d",x,y, x<<y);
}
```

9. (1.0) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
main(){
    int w=1;
    float y=4.5;
    if (y/w) printf("%d\n",(int)(w--?y:5*w));
    w?printf("Test 1\n"):printf("%d\n",(2+w)/++w);
}
```

4
2

10. (1.0) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
#define PODELI(x,y) (x/y)
main() {
    int w=8;
    float y=4,0;
    printf("%.2f\n", PODELI (w + (int)y, w - y));
}
```

4.00

11. (1.5) Šta je rezultat sledećeg koda

```
#include <stdio.h>
main(){
    int x, y, d = 0;
    scanf("%d%d", &x, &y);

    while (x>0)
    {
        y >>= --x/2;
        printf("%d %d\n",x, y);
        if (!y) continue;
        d++;
    }

    printf("%d\n",d); }
```

Ako se kao vrednosti promenljivih x i y unesu:

5 17

12 32

6 31

5

1

2

12. (1.5) Šta je rezultat sledećeg koda?

```
char *str[] = { "februar", "mart", "april", "maj"};
char **s1 = str;
int n = sizeof(str)/sizeof(str[0])-1;
while(n--)
printf("%c\n",*(++(*s1++)));
```

e
a
p