

Strukture podataka i algoritmi 1

Test I (maksimalno 13 poena)

Ime i prezime	Indeks	Broj poena
---------------	--------	------------

1. Napisati rezultate sledećih kodova i dati odgovore na pitanja:

a)	<pre>float f = 7.2, c = 3.6; printf("%d, %d, %d\n", (int)(f/c), (int)((int)f/c), (int)((int)f/(int)c));</pre>	2, 1, 2
b)	<pre>int a = 1, b = 3; int *p = &a, *q = &b; *q = *p; p = q; (*p)--; a++; printf("%d, %d", a, b);</pre>	2, 0
c)	<pre>int x = 0, y = 1; printf("%d,%d\n", x++ !--x ? x-- + y : y++ + x, x);</pre>	1, -1
d)	<pre>#define DOUBLE(x) x + x #define SQUARE(x) x * x int main() { int a = 3; int b = 10 / DOUBLE(a); int c = SQUARE(a+1); int d = (b > c) ? b-c : c-b; printf("%d\n", d); return 0; }</pre>	1

2. Dati odgovore na sledeća pitanja:

- a) Koja Linux komanda omogućava brisanje direktorijuma zajedno sa svim njegovim sadržajem?
Napisati primer upotrebe ove komande.

Komanda: **rm -r**

Primer: **rm -r dir1**

- b) Pomoću kojih funkcija je sve moguće dinamički alocirati memoriju u C programskom jeziku?
Objasniti način upotrebe svake funkcije i dati primer.

Malloc: `int *a = malloc(n*sizeof(int))`

Calloc: `int *a = calloc(n, sizeof(int))`

Realloc: `int *a = realloc(a, n * sizeof(int))`

3. Objasniti šta radi sledeća funkcija i koji rezultat vraća ako se pozove: sa fun(0xA7)? Rezultat dati u heksadekadnom ili binarnom formatu.

```
unsigned int fun(unsigned char n)
{
    unsigned char result = 0;
    for (int i = 0; i < 8; i++) {
        result = (result << 1) | (n & 1);
        n >>= 1;
    }
    return result;
}
```

Funkcija invertuje redosled bitova u unetom broju veličine jednog bajta.
Za unetu vrednost vraća: 0xe5

4. Napisati rezultat izvršavanja sledećeg koda:

```
#include <stdio.h>
int *func(int *a, int n, int t) {
    for (int i = 0; i < n; i++) {
        if (*(a + i) == t) {
            *(a + i) = 0;
            return a + i;
        }
    }
    return NULL;
}
int main() {
    int a[] = {5, 3, 8, 3, 9};
    int *p = func(a, 5, 3);
    if (p != NULL) printf("%ld %d\n", p - a, *p);
    for (int i = 0; i < 5; printf("%d ", a[i]), i++);

    return 0;
}
```

1 0
5 0 8 3 9

5. Napisati rezultat izvršavanja sledećeg koda:

```
#include <stdio.h>
int a = -1;
void func(int b)
{
    static int a = 0;
    a += b;
    if (!a++)
        printf("%d ", a);
    else
    {
        int a = 2;
        printf("%d ", a);
    }
}

int main()
{
    printf("%d ", a);
    int a = 0;
    printf("%d ", a);
    func(a--);
    func(--a);
}
```

-1 0 1 2

6. Napisati funkciju koja prihvata dva cela broja x i n , i vraća broj koji se dobija tako što se u binarnoj reprezentaciji broja x invertuju (menjaju 0 u 1 i 1 u 0) svi oni bitovi koji se nalaze na pozicijama na kojima broj n ima vrednost 1 u binarnoj reprezentaciji. Funkciju implementirati korišćenjem bitovskih operacija.

```
int func(int x, int n)
{
    return x ^ n;
}
```

7. Napisati **rekurzivnu** funkciju koja za dati ceo broj n kao rezultat vraća broj parnih cifara u tom broju. Funkciji nije dozvoljeno prosleđivati dodatne parametre osim celog broja n .

```
int broj_parnih_cifara(int n)
{
    if (n == 0) return 0;
    return (n % 2 == 0 ? 1 : 0) + broj_parnih_cifara(n / 10);
}
```

8. Napisati funkciju "*kreiraj*" koja prihvata ceo broj n i dinamički kreira niz od n elemenata celobrojnog tipa. U glavnom programu učitati jedan ceo broj (broj elemenata niza), kreirati niz koristeći datu funkciju a zatim uneti elemente niza sa standardnog ulaza. Napisati još jednu funkciju, "*modifikuj*", koja prihvata niz celih brojeva i dužinu niza, a zatim modifikuje niz tako da svaki element postaje zbir svih elemenata koji se nalaze levo od njega u nizu.

Primer: Nakon sto se prosledi funkciji "modifikuj":

Niz [6 4 2 3 6 2 1 1] postaje [0 6 10 12 15 21 23 24].

```
int* kreiraj(int n) {
    int *niz = (int*) malloc(n * sizeof(int));
    if (niz == NULL) {
        printf("Greska pri alokaciji memorije!\n");
        exit(1);
    }
    return niz;
}

void modifikuj(int *niz, int n) {
    int zbir = 0;
    for (int i = 0; i < n; i++) {
        int temp = niz[i];
        niz[i] = zbir;
        zbir += temp;
    }
}

int main() {
    int n;

    printf("Unesite broj elemenata niza: ");
    scanf("%d", &n);

    int *niz = kreiraj(n);

    printf("Unesite %d elemenata:\n", n);
    for (int i = 0; i < n; i++) {
        printf("niz[%d] = ", i);
        scanf("%d", &niz[i]);
    }

    modifikuj(niz, n);

    free(niz);
    return 0;
}
```