

# Strukture podataka i algoritmi 1

## I test – max 13 poena

11.04.2024.

Ime i prezime	Broj indeksa	Broj poena

1. Napisati rezultate sledećih kodova (2 poena, svaki primer po 0.5 poena):

a)

```
int a = 10;
int* pa, b;
b = &a;
pa = b;
*pa = 20;
printf("%d %d %d\n", a, *pa, *b);
```

Kompajlerska greška. b je tipa int a ne int\*.

c)

```
#define IZRAZ (x) (2 * x - 3)

int main() {
    int x = 10;
    printf("%d\n", IZRAZ(20 + x));
    return 0;
}
```

47

b)

```
printf('r');
```

Kompajlerska greška. Prosleđen je char umesto stringa.

d)

```
int a = 10, b = 20;
int *pa = &a, *pb = &a;
*pa = a + *pb;
*pb = b;
*pb = *pa + 3;
printf("%d %d %d %d\n", a, b, *pa, *pb);
```

23 20 23 23

2. a) Objasniti šta je terminal i navesti makar 3 Linux komande, koje možemo zadati terminalu. (0.5 poena)

**Terminal** je program koji omogućava korisniku interakciju sa operativnim sistemom putem tekstualnog interfejsa. Komanda: man, ls, cd

b) Objasniti šta rade komande **man** i **gcc**? (0.5 poena)

**man** komanda se koristi kako bi se dobila dokumentacija tj. pomoć o različitim komandama i sistemskim funkcijama. **gcc** komanda služi za kompajliranje .c fajlova i kreiranje izvršnog fajla.

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

```
#include <stdio.h>

int bSearch(int arr[], int l, int r, int x) {
    if (r >= l) {
        int mid = l + (r - l) / 2;

        if(arr[mid] == x)
            return mid;

        if(arr[mid] > x)
            return bSearch(arr, l, mid - 1, x);

        return bSearch(arr, mid + 1, r, x);
    }

    return -1;
}

int main() {
    int arr[] = { 2, 3, 4, 10, 40 };
    int x = 10;
    int n = sizeof(arr) / sizeof(arr[0]);
    printf("%d\n", bSearch(arr, 0, n - 1, x));
    return 0;
}
```

3

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

```
#include <stdio.h>

int main() {
    int x = 12, y = 7, z = 3;
    x |= y;
    z &= x;
    y ^= z;
    int nx = ~z;
    printf("x = %d, y = %d, z = %d, nx = %d\n", x, y, z, nx);
    return 0;
}
```

```
}
```

x = 15, y = 4, z = 3, nx = -4

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

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

```
int f(int a) {  
    static int b = 8;  
    b++;  
    int c = (a++ > ++b)? a : (--b < ++a)? b : 100;  
    b += 5;  
    return c;  
}
```

```
int main() {  
    printf("%d\n", f(10));  
    printf("%d\n", f(15));  
    printf("%d\n", f(20));  
  
    return 0;  
}
```

9

15

21

6. Napisati program koji računa prosek prethodno unetog niza celih brojeva. Program implementirati bez korišćenja operatora [] prilikom unosa članova niza i uzimanja vrednosti članova niza. Niz je potrebno dinamički alocirati. Smatrati da se dužina niza i članovi niza unose sa tastature. (1.5 poen)

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

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

```
int main() {  
    int n;  
  
    scanf("%d", &n);  
  
    int *niz = (int*) malloc(n * sizeof(int));  
  
    int suma = 0;
```

```

for (int i = 0; i < n; i++) {
    scanf("%d", niz + i);
    suma += *(niz + i);
}

printf("%lf\n", 1.0 * suma / n);

return 0;
}

```

7. Napisati funkciju koja proverava da li je broj stepen broja 2 koristeći bitovske operatore. (1.5 poen) Primer poziva:  $f(16) = 1$ ,  $f(32) = 1$ ,  $f(13) = 0$ ,  $f(12) = 0$

```

int f(int n) {
    unsigned int mask = sizeof(int) * 8;

    int br = 0;

    for (mask = 1 << (mask - 1); mask; mask >>= 1) {
        if (n & mask) {
            br++;

            if (br > 1) {
                return 0;
            }
        }
    }

    return 1;
}

```

8. Napisati program kojim se štampaju svi trocifreni brojevi koji su stepen broja k. Broj k se unosi sa standardnog ulaza. Napomena: Provera da li je neki broj stepen broja k vrši funkcija **proveri**, koja ima sledeći potpis **int proveru(int n, int k)**, koja **mora** biti **rekurzivna** i vraća 1 ukoliko je broj n stepen broja k ili 0 ukoliko n nije stepen broja k. (2 poena). Primer poziva:  $proveri(16, 2) = 1$ ,  $proveri(32, 6) = 0$

```

#include <stdio.h>

int proveru(int n, int k) {
    if (n == 1)
        return 1;

    if (n % k != 0)
        return 0;

    return proveru(n / k, k);
}

```

```

}

int main() {
    int k;

    scanf("%d", &k);

    for (int i = 100; i < 1000; i++) {
        if (proveri(i, k))
            printf("%d\n", i);
    }

    return 0;
}

```

9. Napisati funkciju koja proverava da li broj zapisan u binarnom zapisu ima isti broj jedinica i nula. Nije potrebno brojati nule nakon jedinice sa najvećom težinom. (2 poena) Primer poziva:  $f(16) = 0$ ,  $f(10) = 1$

```

int f(int n) {
    unsigned int mask = sizeof(int) * 8;

    int brJedinica = 0;
    int brNula = 0;
    int flag = 0;

    for (mask = 1 << (mask - 1); mask; mask >>= 1) {
        if (flag) {
            if (n & mask)
                brJedinica++;
            else
                brNula++;
        } else {
            if (n & mask) {
                flag = 1;
                brJedinica++;
            }
        }
    }

    return brJedinica == brNula;
}

```