

Strukture podataka i algoritmi 1  
I test – max 13 poena

11.4.2022.

Ime i prezime	Broj indeksa	Broj poena

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

a)

```
int *x;
int a=15;
x=&a;
*x=*x+3;
printf("x=%d, a=%d\n",*x, a);
```

x=18 a=18

b)

```
putchar(35+30);
```

ascii vrednost broja 65  
'A'

c)

```
#define T(x) (x - x / 2)

int main() {
    int x = 6;
    printf("%d", T(x + 2));
}
```

3

d)

```
int a = 5;
int *p = &a;
int b = 3;
p = &b;
*p = *p + a;
a = b - a;
printf("a=%d, b=%d, *p=%d\n",
a, b, *p);
```

a=3 b=8 \*p=8

2. Zaokruži slovo (slova) ispred tačnih tvrdnji (1 poen):

- a) gcc -c zadatak.c; gcc -o zadatak zadatak.c; ./zadatak  
b) gcc -c zadatak.c; gcc -o zadatak zadatak.o; ./zadatak  
c) gcc -c zadatak zadatak.c; ./zadatak  
d) gcc -o zadatak zadatak.c; ./zadatak  
e) gcc -c zadatak zadatak.c; gcc -o zadatak zadatak.c; ./zadatak

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

```
#include<stdio.h>

void fun(int arr[], int start, int end)
{
    if (start >= end) {
        return;
    } else {
        int temp = arr[start];
        arr[start] = arr[end];
        arr[end] = temp;
        fun(arr, start+1, end-1);
    }
}

main()
{
    int arr[] = {2, 7, 8, 1, 5, 9, 3, 7};
    int size = sizeof(arr)/sizeof(arr[0]);
    fun(arr, 2, 6);
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }
}
```

2 7 3 9 5 1 8 7

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

```
#include<stdio.h>

int main() {
    int a = 3, b = 4, c = a < b || a++ < b++ ? a++ : --b;
    printf("a=%d, b=%d, c=%d\n", a, b, c);
}
```

a=4, b=4, c=3

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

```
#include <stdio.h>
int fun()
{
    static int a = 10;
    static int b = 10;
    if(++a>=b++)
        return ++b;
    else
        return a;
}
main()
{
    printf("%d\n",fun());
    printf("%d\n",fun());
}
```

12

14

6. Napisati funkciju koja transformiše zadati broj **N** u niz jedinica i nula koji predstavlja binarni zapis tog broja. U glavnom delu programa je potrebno učitati dva prirodna broja, transformisati ih u niz nula i jedinica koristeći prethodno kreiranu funkciju i sabrati binarne reprezentacije unetih brojeva. Odštampati rezultat sabiranja u binarnom obliku.

Npr. **16 (0 0 0 1 0 0 0 0)** i **7 (0 0 0 0 0 0 1 1 1)** => **(0 0 0 1 0 1 1 1)**.

**Napomena:** neophodno je koristiti bitovske operacije pri kreiranju binarne reprezentacije broja. (2.5 poena)

```
#include <stdio.h>
void array_of_bits(int *arr, int x)
{
    int wl = sizeof(int)*8, i=0;
    unsigned mask;
    for (mask = 1<<wl-1; mask; mask >>=1)
        if (x&mask) arr[i++] = 1;
        else arr[i++] = 0;
}
int main()
{
    int num1[sizeof(int)*8];
    int num2[sizeof(int)*8];
    int num3[sizeof(int)*8];

    int x,y,i;
    scanf("%d%d",&x,&y);

    array_of_bits (num1, x);
    array_of_bits (num2, y);

    int prenos = 0;
    for(i = sizeof(int)*8-1; i >= 0;--i)
    {
        num3[i] = (num1[i] + num2[i] + prenos) % 2;
        prenos = (num1[i] + num2[i] + prenos) / 2;
    }
}
```

7. Napisati program koji, korišćenjem bitovskih operacija, transformiše ceo broj **x** u ceo broj **y**. Brojevi **x** i **y** se unose sa standardnog ulaza. (2.5 poena)

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

8. Napisati rekurzivnu funkciju koja okreće cifre celog broja. Npr. **N=3572 => 2753**.  
**Napomena:** nije dozvoljeno štampati cifre u rekurzivnoj funkciji. (2 poena)

```
#include<stdio.h>
int okreni(int n)
{
    static int pow=10;
    if (n < 10)
    {
        return n;
    }
    else
    {
        int broj = okreni(n/10);
        broj += (n%10)*pow;
        pow*=10;
        return broj;
    }
}
int main()
{
    int n;
    scanf("%d", &n);
    printf("%d", okreni(n));

    return 1;
}
```