

Ime i prezime

Broj indeksa

1. (0.5) Zaokružiti slova ispred tvrdnje (tvrdnji) koje su tačne:

- a) kod sa uključenom bibliotekom **math.h** se kompajlira uz dodavanje argumenata **-l -m**
- b) **kod sa uključenom bibliotekom math.h se kompajlira uz dodavanje argumenta -lm**
- c) u programskom jeziku C ne postoji biblioteka **math.h**
- d) kod sa uključenom bibliotekom **math.h** se kompajlira bez dodatnih argumenata

2. (0.5) Šta je rezultat sledećeg koda

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

main()
{
    float x = 0.32, y = 3.29;
    printf("A");
    if(floor(x))
    {
        if(ceil(y) == 3)
            printf("\n");
        printf("B");
    }
    else if((floor(y) - ceil(x)) == 2)
        printf("C");
    else
        printf("D");
}
```

AC

3. (1.0) Šta se ispisuje u svakom od segmenata na programskom jeziku C? Smatrati da su sve promenljive tipa int i da su ispravno deklarisanе.

- a) for(a=15, b=21, i=15; a%i!=0 || b%i!=0; i--); printf("%d",i); 3
- b) a = 15; b = 21; ind = 0; i = 22; while(!ind) { i--; ind = (a%i==0 && b%i==0)? 1 : 0; } printf("%d",i); 3
- c) a = 15; b = 21; i = 22; do{i--;}while(a%i==0 && b%i==0); printf("%d",i); 21

4. (1.0) Šta je rezultat sledećeg koda

```
#include <stdio.h>

main()
{
    int a = 32, b = 2;
    while(a ^ b)
    {
        a >>= 1;
        b << 1;
        printf("%d\n", a | b);
    }
}
```

18
10
6
2

5. (1.0) Napisati deo koda u kome se datom pozitivnom celom broju x sa desne strane cifre stotina dodaje nula ako je x bar trocifren, a ako nije uvećati ga za 100. Zadatak uraditi na dva načina:

a) korišćenjem samo uslovnog operatora

b) korišćenjem if naredbe

```
x = x >= 100 ? (x / 100) * 1000 + x % 100 : x + 100;      if(x >= 100)
                                                           x = (x / 100) * 1000 + x % 100;
                                                           else
                                                           x = x + 100;
```

6. (1.0) Šta je rezultat sledećeg koda:

```
#include<stdio.h>

int a = 4;                                          3

void funkcija2(int b)                              2
{
    if(a - b)
        printf("%d\n", --a);
    else
        printf("%d\n", b);
}

void funkcija1(int b)
{
    static int a = 3;

    if(++a == b--)
        funkcija2(a--);
    else
        funkcija2(++b);
    a--;
}

main()
{
    funkcija1(5);
    funkcija1(2);
}
```

7. (1.0) Šta je rezultat sledećeg koda

```
#include <stdio.h>

#define M(x, y) x*y + x/y                          1

main()
{
    int k = M(2-5, 2);

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

8. (1.0) Šta je rezultat sledećeg koda

```
#include <stdio.h>

void saberi(const int a[], int b[], int n)
{
    int i;

    for(i = 0; i < n; i++)
        b[i] = b[i] + a[i];
}

main()
{
    int a[] = {1, 2, 3, 4, 5};
    int b[] = {-1, -2, -3, -4, -5};
    int n = 5, i;

    saberi(a, b, n);

    for(i = 0; i < n; i++)
        printf("%d\n", b[i]);
}
```

9.(1.0) Napisati program u programskom jeziku C koji sa ulaza učitava realan broj **m** koji se nalazi u intervalu (0,1) i čije su cifre uređene u nerastućem poretku (**m** sadrži šest cifara posle decimalnog zareza) i ceo broj **n** koji predstavlja cifru koja se umeće u broj **m** tako da se ne naruši poredak. Dozvoljeno je korišćenje nizova.

Primer – za dati ulaz: 0.433211 6 0.864421 3	program ispisuje: 0.6433211 0.8644321
--	---

10. (1.0) Napisati funkciju DeljivSa4 koja za dati ceo broj vraća 1 ukoliko je on deljiv sa 4, a u suprotnom vraća 0. Funkciju implementirati koristeći samo bitovske operacije.

```
int DeljivSa4(int x)
{
    if (x & 3) return 0;
    else return 1;
}
```

9.

```
#include<stdio.h>

int main()
{
    double m;
    int i, n, cifra, mesto;
    int cifre[7]; //6 cifara iz broja + 1 koja se dodaje

    scanf("%lf%d", &m, &n);

    //premestanje cifara u niz
    for(i = 0; i < 6; i++)
    {
        cifra = (int)(floor(m * 10));
        m = 10 * m - cifra;
        cifre[i] = cifra;
    }

    i = 0;

    //trazenje mesta za n
    while(cifre[i] > n && i < 6)
        i++;

    mesto = i;

    //ubacivanje n
    for(i = 6; i > mesto; i--)
        cifre[i] = cifre[i - 1];

    cifre[mesto] = n;

    //nazad u broj
    m = 0.0;

    for(i = 6; i >= 0; i--)
    {
        m += cifre[i];
        m /= 10.0;
    }

    printf("m = %lf\n", m);
}
```