

ZADACI

Višestruko pridruživanje i skraćeno pridruživanje

1. zadatak

Koliko iznose vrijednosti varijabli *a*, *b* i *c* nakon obavljanja programskog odsječka?

```
int a, b, c;  
a = b = c = 2;  
b += ++a + 4;  
b -= a--;  
c += a % b;
```

2. zadatak

Koliko iznosi vrijednosti varijable **suma** nakon obavljanja programskog odsječka?

```
int i, j, suma;  
suma = 0;  
for (i = 3, j = 1; i > 0 || j < 5; i --, j ++) {  
    suma += i + j;  
}
```

Odvajanje naredbi, uvjetno pridruživanje

1. zadatak

Koliko iznose vrijednosti varijabli *a*, *b* i *c* nakon obavljanja programskog odsječka?

```
int a, b, c;  
a = 10;  
c = b = a % 4 ? a + 1 : a;
```

2. zadatak

Pomoću uvjetnog operatora napišite izraz koji će varijabli **c** pridružiti vrijednost 1 ako je u varijablu **a** spremljen ASCII kôd malog ili velikog slova engleske abecede, u suprotnom varijabli **c** pridružiti vrijednost 0.

Programska petlja while, kratki i dugi oblik naredbe

1. zadatak

Što će ispisati sljedeći programski odsječak?

```
int i = 1, j = 8;  
while (i < j) {  
    printf("%d ", i + j);  
    i ++;  
    j --;  
}
```

2. zadatak

Napišite programski odsječak, koristeći *while* petlju, koji izračunava sumu prvih 10 parnih prirodnih brojeva!

Programska petlja do-while, kratki i dugi oblik naredbe

1. zadatak

Što će ispisati sljedeći programski odsječak?

```
int i = 1, j = 1;
do
{
    printf("%d %d\n", i ,j);
    i += j;
    j = j * 2 % 3;
} while (j + i < 5 || i < 10);
```

2. zadatak

Napišite programski odsječak, koristeći dvije *do-while* petlje, koji ispisuje sva velika slova engleske abecede na način da se svako slovo ispiše onoliko puta koliko iznosi njegov redni broj u abecedi.

Početak ispisa treba izgledati ovako:

```
A
BB
CCC
DDDD
EEEE
....
```

Programska petlja for

1. zadatak

Što će ispisati sljedeći program?

```
#include <stdio.h>
int main() {
    int i, n, suma;
    float arit_sred;

    suma = 0;
    n = 10;
    for(i = 0; i < n; i ++) {
        suma += i;
    }
    arit_sred = (float) suma / n;
    printf("Aritmeticka sredina je %f\n", arit_sred);
    return 0;
}
```

2. zadatak

Što će ispisati sljedeći program?

```
#include <stdio.h>
int main() {
    float m = 2, i;

    for(i = 0; i <= m; i += 0.5 ) {
        printf("%f\n", i);
    }
    return 0;
}
```

Naredba switch

1. zadatak

Što će se na zaslonu ispisati sljedećim programom?

```
int main () {
    char c;
    c = '1' + 1;
    switch (c) {
        case '1': printf ("%d", c);
        case '2': printf ("%d", c);
        case '3': printf ("%d", c); break;
        default: printf ("default");
    }
    return 0;
}
```

2. zadatak

Napišite program kojim ćete učitati redni broj dana u tjednu i ispisati naziv učitano dana. U zadatku koristiti naredbu switch.

Korištenje goto

1. zadatak

Izvođenje programskog odsječka:

```
while(i < 10) {
    if ((a-i)%4 == 0) goto lab1;
    sum += i;
lab1:
    i++;
}
```

ostalo bi nepromijenjeno ako bi se naredbu `goto lab1;` zamijenilo s

- a) naredbom `continue;`
- b) naredbom `break;`
- c) naredbom `printf ("\n");`
- d) naredbom `stop;`
- e) niti jedan od ponuđenih odgovora nije točan

2. zadatak

Što će se ispisati sljedećim programskim odsječkom ?

```
int main() {
    int i=0;
    ponovo:
        printf("%d ", ++i );
        if(i++ <= 5 ) goto ponovo;

    return 0;
}
```

3. zadatak

Izvođenje programskog odsječka:

```
while(i < 10) {
    i++;
    if ((a-i)%4 == 0) goto lab1;
    sum += i;
    lab1:
        ;
}
```

ostalo bi nepromijenjeno ako bi se naredbu `goto lab1;` zamijenilo s

- a) naredbom `continue;`
- b) naredbom `break;`
- c) naredbom `printf ("\n");`
- d) naredbom `stop;`
- e) niti jedan od ponuđenih odgovora nije točan

break, continue

1. zadatak

Što će ispisati sljedeći programski odsječak?

```
int sum = 0, i, j, gotovo = 0;
for( i = 1; i <= 5 && !gotovo; i++){
    for( j = 1; j <= 5 ; j++){
        if (j%4 == 0){
            gotovo = 1;
            continue;
        }
        sum += i+j;
    }
}
```

2. zadatak

Što će se ispisati nakon obavljanja programskog odsječka:

```
int i, j;
for (i = 1; i <= 3; i++)
```

```
for (j = 1; j <= 4; j++) {  
    if (j%2) continue;  
    printf(" %d ", j);  
    if (j!=i) break;  
}
```

Korištenje bitovnih operatora

1. zadatak

Koja je vrijednost varijable a nakon izvođenja sljedećeg programskog odsječka:

```
signed char a=5;  
a=~a;
```

2. zadatak

Koja je vrijednost u varijabli c nakon izvođenja sljedećeg programskog odsječka :

```
int a=6, b=12, c;  
c = (a & b) + (a | b) + (a ^ b);
```

3. zadatak

Što će ispisati sljedeći program:

```
#include <stdio.h>  
int main(){  
    char a,b,c;  
    b = 1;  
    a = b << 2 < 2;  
    c = a ^ b;  
    printf("a=%d, b=%d, c=%d",a,b,c);  
    return 0;  
}
```

Degenerirane programske petlje (beskonačne, one koje se ne izvršavaju)

1. zadatak

Što će se ispisati nakon izvođenja sljedećeg programa:

```
int main () {  
    int i;  
    i = 0;  
    do {  
        ++i;  
        if (i%2 == 1){  
            i *= 2;  
        } else {  
            i /= 2;  
        }  
        printf("%d",i);  
    } while (i = 1);  
    return 0;  
}
```

2. zadatak

Što će se ispisati izvođenjem sljedećeg programa:

```
int main () {  
    int i, j;  
    for (i=0; i==i+1; ++i){  
        for (j=0; j<2; j++){  
            printf("*");  
        }  
    }  
    return 0;}
```

Kombinacije više tipova petlji

1. zadatak

Što će se ispisati nakon izvođenja sljedećeg programa?

```
int main()
{
    int i = 0;
    int suma=0;

    while(i < 10)
    {
        for(i = 0; i < 10; i++)
        {
            suma += 1;
        }
        i++;
    }
    printf("Suma = %d", suma);
    return 0;
}
```

2. zadatak

Što će se ispisati nakon izvođenja sljedećeg programa?

```
int main()
{
    int i = 0;
    int j = 0;
    int suma=0;

    while(i < 10)
    {
        for(j = 0; j < 10; j++)
        {
            suma += 1;
        }
        i++;
    }
    printf("Suma = %d", suma);
    return 0;
}
```

Kombinacije selekcija i programskih petlji

1. zadatak

Što radi sljedeći programski odsječak?

```

int main()
{
    int i = 2;
    int ucitaniBroj = 0;
    int suma = 0;

    scanf("%d", &ucitaniBroj);
    while(i < ucitaniBroj)
    {
        if(!(i%2)) suma += i;
        i++;
    }
    printf("%d", suma);
    return 0;
}

```

2. zadatak

U ovome je zadatku programski odsječak iz 1. zadatka napisan na malo drukčiji način (pomoću do – while petlje). Za negativne vrijednosti varijable ucitaniBroj će se, za razliku od prethodnog primjera, izvršiti jedan prolaz kroz do-while petlju. Odgovoriti koliku vrijednost poprima varijabla suma za ucitaniBroj=-4 te komentirati rezultat u svjetlu uvodnog teksta.

```

int main()
{
    int i = 2;
    int ucitaniBroj = 0;
    int suma = 0;

    scanf("%d", &ucitaniBroj);
    do
    {
        if(!(i%2)) suma += i;
        i++;
    } while(i < ucitaniBroj);
    printf("%d", suma);
    return 0;
}

```

RJEŠENJA ZADATAKA

Višestruko pridruživanje i skraćeno pridruživanje

1. zadatak

a=2, b=6, c=4

2. zadatak

suma=16

Odvajanje naredbi, uvjetno pridruživanje

1. zadatak

a=10 b=11 c=11

2. zadatak

```
c = a >='A' && a <='Z' || a >='a' && a <='z' ? 1:0;
```

Programska petlja while, kratki i dugi oblik naredbe

1. zadatak

9 9 9 9

2. zadatak

```
int suma = 0, i = 1;
while (i <= 10) {
    suma += 2*i;
    i++;
}
```

Programska petlja do-while, kratki i dugi oblik naredbe

1. Zadatak

```
1 1
2 2
4 1
5 2
7 1
8 2
```

2. Zadatak

```
char c='A';int i;
do
{
    i=0;
    do
    {
        printf("%c",c);
    }while(i++ < c-'A');
    printf("\n");
}while(c++ != 'Z');
```

Programska petlja for

1. zadatak

4.500000

2. zadatak

0.000000

0.500000

1.000000

1.500000

2.000000

Naredba switch

1. zadatak

5050

2. zadatak

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

```
int main ()
```

```
{
```

```
    int dan;
```

```
    printf ("Unesite redni broj dana");
```

```
    scanf ("%d", &dan);
```

```
    switch (dan){
```

```
        case(1): printf ("Ponedjeljak"); break;
```

```
        case(2): printf ("Utorak"); break;
```

```
        case(3): printf ("Srijeda"); break;
```

```
        case(4): printf ("Cetvrtak"); break;
```

```
        case(5): printf ("Petak"); break;
```

```
        case(6): printf ("Subota"); break;
```

```
        case(7): printf ("Nedjelja"); break;
```

```
        default: printf ("Neispravno unesen dan.");
```

```
    }
```

```
    return 0;
```

```
}
```

Korištenje goto

1. zadatak

e) niti jedan od ponuđenih odgovora nije točan

2. zadatak

1 3 5 7

3. zadatak

a) naredbom continue;

break, continue

1. zadatak

15

2. zadatak

2 2 4 2

Korištenje bitovnih operatora

1. zadatak

-6

2. zadatak

28

3. zadatak

a=0, b=1, c=1

Degenerirane programske petlje (beskonačne, one koje se ne izvršavaju)

1. zadatak

Petlja će se obaviti beskonačno zato jer se prilikom ispitivanja uvjeta petlje:

```
while (i = 1)
```

koristi naredba za pridruživanje (=), a ne uspoređivanje (==) te se uvjet uvijek izračunava kao 1 (istina).

2. zadatak

Ništa, zato jer je u uvjetu petlje $i == i + 1$ a taj izraz nije nikad istinit pa se petlja neće obaviti niti jednom.

Kombinacije više tipova petlji

1. zadatak

Ispisat će se 10, a ne 100 kao što bi se moglo inicijalno očekivati. Naime, zbog korištenja varijable i i u vanjskoj i u unutarnjoj petlji, vanjska petlja će se izvršiti samo jednom. Za ispravan rad bi trebalo varijablu i u unutarnjoj petlji zamijeniti s novom varijablom (npr j)

2. zadatak

Ispisat će se 100. Za razliku od 1. zadatka pazilo se kod korištenja varijabli – svaka je petlja dobila "svoju" varijablu čime je riješen "problem" iz prethodnog zadatka, ali i olakšana čitljivost programa (sada je i osobi koja prije nije vidjela program sve mnogo razumljivije).

Kombinacije selekcija i programskih petlji

1. zadatak

Programski odsječak sumira sve parne brojeve između nule i broja ucitaniBroj te ispisuje dobiveni rezultat – npr za ucitaniBroj=5 ispisuje 6.

2. zadatak

Programski odsječak radi istu stvar kao i programski odsječak u prvome zadatku, ali zbog korištenja do-while petlje uvijek ima jedan prolazak kroz nju. Baš će zbog toga, ukoliko se unese negativan parni broj u varijablu ucitaniBroj (npr. -4), doći do neželjenih rezultata – obaviti će se jedan prolaz kroz do-while petlju i ispisati broj različit od nule kao vrijednost sume (2). To se ne bi smjelo dogoditi budući da se traži suma parnih brojeva između 2 i -4 (donja granica je veća od gornje, pa takvi brojevi ne postoje).

Iz ovoga je primjera vidljivo da se petlje s ispitivanjem uvjeta na početku i petlje s ispitivanjem uvjeta na kraju trebaju koristiti s oprezom i ovisno o tome što u konačnici želimo postići.