

## 17. Dodatne vježbe

U svim zadacima u kojima se traži definiranje funkcije, treba napisati odgovarajući glavni program (tj. funkciju `main`) u kojem ćete po potrebi definirati stvarne argumente, pozvati funkciju i ispisati rezultat.

1. Što će se ispisati tijekom izvođenja sljedećeg programa:

```
#include <stdio.h>

void fun(void) {
    int x = 5;
    printf("%d\n", x);
    x++;
}

int main () {
    fun();
    fun();
    return 0;
}
```

2. Što će se ispisati tijekom izvođenja sljedećeg programa:

```
#include <stdio.h>

void fun(void) {
    static int x = 5;
    printf("%d\n", x);
    x++;
}

int main () {
    fun();
    fun();
    return 0;
}
```

3. Što će se ispisati tijekom izvođenja sljedećeg programa:

```
#include <stdio.h>

void fun(void) {
    static int x = 5;
    int y = 5;
    printf("%d %d\n", ++x, --y);
}

int main () {
    fun();
    fun();
    fun();
    return 0;
}
```

4. Što će se ispisati tijekom izvođenja sljedećeg programa:

```
#include <stdio.h>

int main () {
    static int i = 5;
    int prviPut = 1;
labela:
    {
        static int i = 10;
        int j = 15;
        printf("%d %d\n", i, j);
        i++;
        j++;
    }
    i++;
    printf("%d\n", i);
    if (prviPut) {
        prviPut = 0;
        goto labela;
    }
    return 0;
}
```

5. Što će se ispisati tijekom izvođenja sljedećeg programa:

```
#include <stdio.h>

static int x = 25;

void fun1(void) {
    static int x = 5;
    printf("%d\n", ++x);
}

void fun2(void) {
    int x = 10;
    printf("%d\n", ++x);
}

void fun3(void) {
    printf("%d\n", ++x);
}

int main () {
    x++;
    {
        static int x = 15;
        {
            int x = 20;
            printf("%d\n", x++);
        }
        printf("%d\n", x++);
    }
    printf("%d\n", x++);
    fun1();
    fun2();
    fun3();

    fun1();
    fun2();
    fun3();
    return 0;
}
```

6. Napisati funkciju `zbroji` tipa `int` koja vraća zbroj dvaju zadanih cijelih brojeva i funkciju `mnozi` tipa `int` koja vraća umnožak dvaju zadanih cijelih brojeva. Svaka od funkcija, osim što izračunava rezultat i vraća ga u pozivajući program, na zaslon ispisuje koliko je puta bila pozvana. Npr. ako se u glavnom programu obave naredbe:

```
printf("2*2=%d\n", mnozi(2,2));
printf("2+3=%d\n", zbroji(2,3));
printf("4+2=%d\n", zbroji(4,2));
printf("2*5=%d\n", mnozi(2,5));
printf("2*3=%d\n", mnozi(2,3));
```

na zaslonu se treba ispisati:

Funkcija mnozi do sada je pozvana 1 puta	← ispisano u funkciji <code>mnozi</code>
2*2=4	← ispisano u glavnom programu
Funkcija zbroji do sada je pozvana 1 puta	← ispisano u funkciji <code>zbroji</code>
2+3=5	← ispisano u glavnom programu
Funkcija zbroji do sada je pozvana 2 puta	← ispisano u funkciji <code>zbroji</code>
4+2=6	← ispisano u glavnom programu
Funkcija mnozi do sada je pozvana 2 puta	← ispisano u funkciji <code>mnozi</code>
2*5=10	← ispisano u glavnom programu
Funkcija mnozi do sada je pozvana 3 puta	← ispisano u funkciji <code>mnozi</code>
2*3=6	← ispisano u glavnom programu

7. Slično kao prethodni zadatak, uz dodatak: svaka od funkcija mora ispisati ne samo koliko je puta bila pozvana ona sama, nego i koliko puta je bila pozvana bilo koja od funkcija `zbroji` i `mnozi`. Npr. ako se u glavnom programu obave naredbe:

```
printf("2*2=%d\n", mnozi(2,2));
printf("2+3=%d\n", zbroji(2,3));
printf("4+2=%d\n", zbroji(4,2));
printf("2*5=%d\n", mnozi(2,5));
printf("2*3=%d\n", mnozi(2,3));
```

na zaslonu se treba ispisati:

Funkcija mnozi do sada je pozvana 1 puta	← ispisano u funkciji <code>mnozi</code>
Funkcije zbroji i mnozi do sada su pozvane 1 puta	← ispisano u funkciji <code>mnozi</code>
2*2=4	← ispisano u glavnom programu
Funkcija zbroji do sada je pozvana 1 puta	← ispisano u funkciji <code>zbroji</code>
Funkcije zbroji i mnozi do sada su pozvane 2 puta	← ispisano u funkciji <code>zbroji</code>
2+3=5	← ispisano u glavnom programu
Funkcija zbroji do sada je pozvana 2 puta	← ispisano u funkciji <code>zbroji</code>
Funkcije zbroji i mnozi do sada su pozvane 3 puta	← ispisano u funkciji <code>zbroji</code>
4+2=6	← ispisano u glavnom programu
Funkcija mnozi do sada je pozvana 2 puta	← ispisano u funkciji <code>mnozi</code>
Funkcije zbroji i mnozi do sada su pozvane 4 puta	← ispisano u funkciji <code>mnozi</code>
2*5=10	← ispisano u glavnom programu
Funkcija mnozi do sada je pozvana 3 puta	← ispisano u funkciji <code>mnozi</code>
Funkcije zbroji i mnozi do sada su pozvane 5 puta	← ispisano u funkciji <code>mnozi</code>
2*3=6	← ispisano u glavnom programu

**Rješenja svih zadataka provjeriti prevođenjem i testiranjem vlastitih programa!**

## Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

### Rješenje 1. zadatka

Pri svakom pozivu funkcije varijabla x se ponovno inicijalizira. Ispisat će se:

5  
5

### Rješenje 2. zadatka

Varijabla x se inicijalizira samo jednom, na početku izvođenja programa, a njezina vrijednost ostaje sačuvana do kraja izvođenja programa (ne gubi se završetkom funkcije). Ispisat će se:

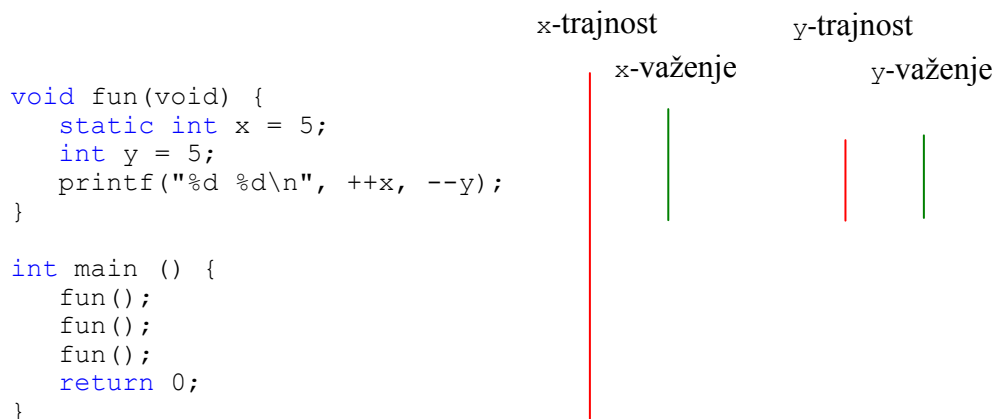
5  
6

### Rješenje 3. zadatka

Potrebno je uočiti koje su varijable definirane u programu, te na temelju smještajnog razreda kojem pripadaju odrediti njihovo područje važenja i trajnost.

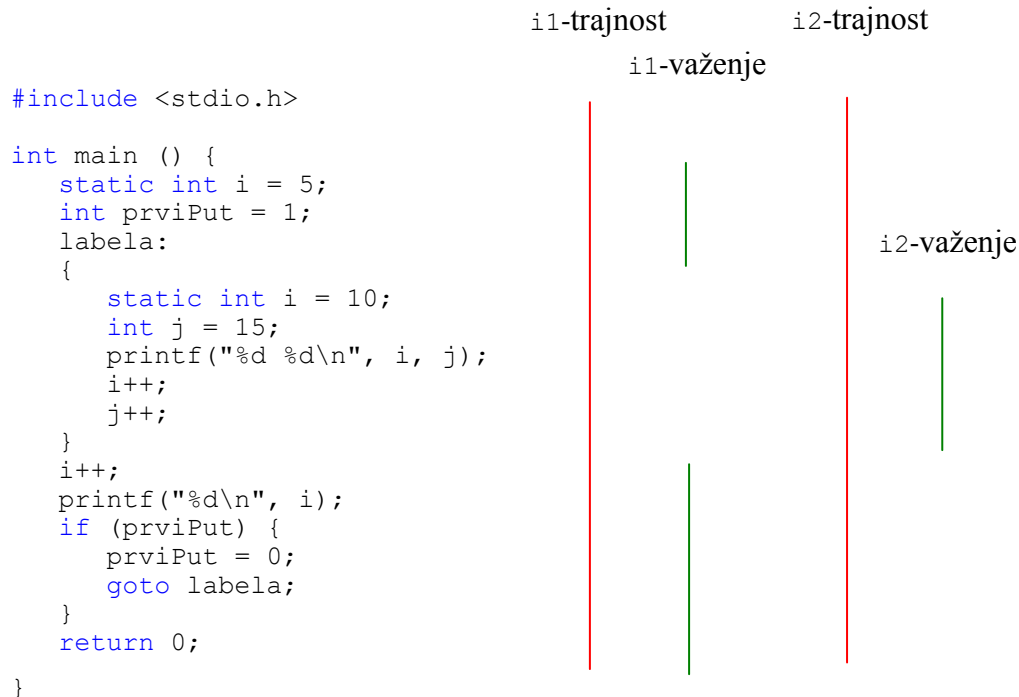
Varijabla x je statička varijabla. To znači da se njezina trajnost proteže od početka do završetka programa. Varijabla se inicijalizira **samo jednom**, na početku izvođenja programa (čak i prije nego se prvi puta pozove funkcija), te njena vrijednost ostaje sačuvana do završetka programa. Varijabla x je definirana unutar funkcije, stoga se njezino područje važenja (tj. "područje programa u kojem je vidljiva") proteže od mjesta u funkciji na kojem je definirana do kraja funkcije.

Varijabla y je automatska varijabla. Varijabla se inicijalizira svaki puta kad se pozove funkcija, a njezina vrijednost se gubi u trenutku završetka funkcije.



## Rješenje 4. zadatka

Ovdje je važno uočiti da postoje dvije varijable naziva `i`. Na slici koja prikazuje područja važenja i trajnosti, prva varijabla `i` (definirana na početku glavnog programa) označena je oznakom `i1`, a druga oznakom `i2`.



## Rješenje 5. zadatka

Ovdje je važno uočiti da postoji 5 različitih varijabli naziva `x`. Kad se odredi trajnost i područje važenja svake od tih varijabli, zadatak je lako riješiti.

## Rješenje 6. zadatka

```
#include <stdio.h>

int zbroji(int x, int y) {
    static int brojPoziva = 0;
    brojPoziva++;
    printf("Funkcija zbroji do sada je pozvana %d puta\n", brojPoziva);
    return x+y;
}

int mnozi(int x, int y) {
    static int brojPoziva = 0;
    brojPoziva++;
    printf("Funkcija mnozi do sada je pozvana %d puta\n", brojPoziva);
    return x*y;
}

int main () {
    printf("2*2=%d\n", mnozi(2,2));
    printf("2+3=%d\n", zbroji(2,3));
    printf("4+2=%d\n", zbroji(4,2));
    printf("2*5=%d\n", mnozi(2,5));
    printf("2*3=%d\n", mnozi(2,3));

    return 0;
}
```

## Rješenje 7. zadatka

```
#include <stdio.h>

int ukupniBrojPoziva = 0;

int zbroji(int x, int y) {
    static int brojPoziva = 0;
    brojPoziva++;
    ukupniBrojPoziva++;
    printf("Funkcija zbroji do sada je pozvana %d puta\n", brojPoziva);
    printf("Funkcije zbroji i mnozi do sada su pozvane %d puta\n", ukupniBrojPoziva);
    return x+y;
}

int mnozi(int x, int y) {
    static int brojPoziva = 0;
    brojPoziva++;
    ukupniBrojPoziva++;
    printf("Funkcija mnozi do sada je pozvana %d puta\n", brojPoziva);
    printf("Funkcije zbroji i mnozi do sada su pozvane %d puta\n", ukupniBrojPoziva);
    return x*y;
}

int main () {
    printf("2*2=%d\n", mnozi(2,2));
    printf("2+3=%d\n", zbroji(2,3));
    printf("4+2=%d\n", zbroji(4,2));
    printf("2*5=%d\n", mnozi(2,5));
    printf("2*3=%d\n", mnozi(2,3));

    return 0;
}
```