

1. [16 T] Kaj izpišejo spodnji deli programov :

```
a) string[] niz = {"Tantadruj"};
   int x;
   x = niz.Length;
   System.Console.WriteLine(x);
```

```
b) string niz = "Tantadruj";
   char y;
   y = niz[5];
   System.Console.WriteLine("V nizu je crka " + y);
```

```
c) string niz1 = "Tantadruj";
   int n = niz1.Length;
   string niz2 = "";
   int i = 0;
   while(i < n) {
       char x = niz1[i];
       niz2 = x + niz2;
       i = i + 1;
   }
   System.Console.WriteLine(niz2);
```

```
d) string niz1 = "Tantadruj";
   int n = niz1.Length;
   string niz2 = "";
   int i = 3;
   while(i < n) {
       niz2 = niz2 + niz1[i];
       i = i + 1;
   }
   niz2 = niz2 + niz1[0] + niz1[1] + niz1[2];
   System.Console.WriteLine(niz2);
```

2. [10 T] Najprej smiselno (na dveh mestih) dopolni program

```
public class TabelaImen {
    public static void Main (string[] args) {
        string[] _____ = {"Mojca", "Katja", "Gašper", "Gabriela", "Tine"};

        System.Console.WriteLine("Dolžina tabele je " + _____ + ".");
        System.Console.WriteLine("Ime v tabeli je " + imena[3] + ".");
    }
}
```

Nato odgovori: Kaj se izpiše, ko program prevedemo in požemo?

3. [5T] Dana je naslednja metoda:

```
public static int Prestej(string niz) {  
    int x;  
    x = niz.Length;  
    return x;  
}
```

Kako je ime metodi? _____

Kakšen je tip rezultat, ki ga metoda vrača? _____

Koliko argumentov sprejme metoda? _____

Kakšni so tipi in imena argumentov? _____

Kaj metoda počne? _____

4. [8T] Kaj izpiše naslednja metoda:

```
public static void Izpis () {  
    string a = "moje bojno polje";  
    System.Console.Write(a.IndexOf("oj"));  
}
```

a) 6

b) 7

c) 1

d) 2

e) nič od navedenega, ampak _____

5. [6T] Katere kombinacije znakov izmed navedenih so **relacijski** operatorji v C#

a. =<

b. !=

c. >=

d. <>

e. =!

f. >

g. =

6. [5T] Dana sta ukaza

```
string[] stavek = new string[10];  
stavek[0] = "Konec se bliza";
```

Kateri ukaz moramo uporabiti, če želimo izpisati dolžino niza "Konec se bliza"? Obkroži vse pravilne odgovore!

a) System.Console.WriteLine(stavek.Length());

b) System.Console.WriteLine(stavek[0].Length());

c) System.Console.WriteLine(stavek.Length);

d) System.Console.WriteLine(stavek[0].Length());

Spodnje programe reši vsakega na svoj list, ki ga oddaj na sredi mape.

7. [15T] Napiši **program**, ki prebere tromestno celo število in ugotovi, ali je število desetice večje od vsote enic in stotic tega števila.

Primeri:

Vpiši število: 234;
V 234 desetice NISO večje kot enice + stotice.

Vpiši število: -164;
V -154 desetice SO večje kot enice + stotice.

Vpiši število: -738;
V -738 desetice NISO večje kot enice + stotice.

Vpiši število: 494;
V 404 desetice SO večje kot enice + stotice.

8. [20T] Sestavi **metodo** `ObrniNiz`, ki dani niz (parameter metode) izpiše obrnjeno.

Primeri:

```
klic ObrniNiz("1. januar 2008"); povzroči izpis  
Niz 1. januar 2008 izpisan obrnjeno je 8002 raunaj .1
```

```
klic ObrniNiz("Lahek izpit"); povzroči izpis  
Niz Lahek izpit izpisan obrnjeno je tipzi kehaL
```

9. [30T] Napiši **metodo**, ki kot parameter dobi dve tabeli celih števil. Vrne naj tabelo, v kateri so tiste vrednosti, ki se pojavijo v drugi tabeli in ne v prvi! Vemo, da so vse vrednosti v obeh tabelah med seboj različne (torej se v isti tabeli število nikoli ne ponovi)

Primeri:

- Če so v prvi tabeli podatki 3, 4, 2, 1 in v drugi 3, 4, 6, 5, naj metoda vrne tabelo s podatki 3 in 4.
- Če so v prvi tabeli podatki 3, 41, 2, 11 in v drugi 3, 11, 41, 2 naj metoda vrne prazno tabelo.
- Če so v prvi tabeli podatki 3, 41, 2, 11 in v drugi 100, 17, 11, 13, 41, 2 naj metoda vrne tabelo s podatki 100, 17 in 13.

10. [40T] Pesem na zgoščenci je predstavljena z objektom razreda `Pesem`:

```
public class Pesem  
{  
    public string naslov;  
    public int minute;  
    public int sekunde;  
    public Pesem(string nasl, int min, int sek)  
    {  
        naslov = nasl; minute = min; sekunde = sek;  
    }  
}
```

Na primer, objekt `new Pesem("Echoes", 15, 24)` predstavlja pesem "Echoes", ki traja 15 minut in 24 sekund.

- Sestavi razred `Zgoscenka`, ki vsebuje naslov zgoščence, ime izvajalca in tabelo pesmi na zgoščenci. Predpostavi, da vse podatke (naslov, izvajalca in tabelo pesmi) obvezno dobiš s konstruktorjem in jih naknadno nikoli ne smeš (moreš) spremeniti. Definiraj ustrezne komponente s pravilnim dostopom in konstruktor.
- Razredu `Zgoscenka` dodaj *objektno* metodo `Dolzina()`, ki vrne skupno dolžino vseh pesmi na zgoščenci, izraženo v sekundah.
- Razredu `Zgoscenka` dodaj *objektno* metodo `PovprecnaDolzina()`, ki vrne povprečno dolžino vseh pesmi na zgoščenci, izraženo v sekundah.

-
11. [30T] Podan imamo razred `Oseba` z objektivnima metodama `VrniIme()` ter `VrniPriimek()` (ki nam vrmeta niza, ki predstavljata ime oz. priimek osebe). Sestavite statični metodi, ki:
- Za dano tabelo tipa `Oseba[]` izpiše priimke vseh oseb, katerih skupna dolžina imena in priimka presega `k` znakov, kjer je `k` parameter metode.
 - V tabeli tipa `Oseba[]`, kjer so osebe urejene po priimkih, poiščite poljubno osebo s priimkom `Novak` (ime torej ni pomembno), ter kot rezultat metode vrnite nov objekt, ki ima podatke enake kot najdena oseba. Če takšna oseba ne obstaja, vrnite `null`.
12. [40T] S pomočjo opisanega razreda `Oseba`, ki poleg zgoraj omenjenih metod pozna še metode za nastavljanje imena (`NastaviIme`) in priimka (`NastaviPriimek`) ter konstruktor `Oseba()`, ki naredi osebo z imenom `Janez` in priimkom `Novak`, sestavite razred `Plavalec`, ki poleg imena in priimka vsebuje še podatke o:
- najljubšem stilu plavanja plavalca (indeks ustreznega stila, ki ustreza vnosu iz spodaj omenjene tabele – če je npr. v tabeli stilov plavanja niz "prsno" na 2. tem mestu v tabeli, in želimo za tega plavalca povedati, da najraje plava prsno, imamo tu shranjeno 2, če je "vseeno" in je niz "vseeno" v tabeli na začetku, pa imamo shranjeno 0)
 - možnih stilih plavanja (tabela nizov) – tabela vsebuje vsaj nize "vseeno", "kravl" in "prsno". Tabela se nikoli ne spreminja!

Vsebuje naj dva konstruktorja:

- privzetega, ki ustvari plavalca `Janeza Novaka`, ki plava `kravl`
- konstruktor s tremi parametri (ime, priimek in stil plavanja), ki ustvari podanega plavalca. Če parametri niso pravilni, naj se ustvari privzeti plavalec.

ter potrebni metodi za branje (vrne naj opisno ime stila iz tabele stilov in ne indeks) in spreminjanje najljubšega stila posameznega plavalca (parameter je ime stila). Če parameter pri tej metodi, ki označuje stil, v tabeli stilov ne obstaja, nastavimo najljubši stil plavanja na "vseeno" (torej shranimo podatek 0). Vsebuje naj tudi objektno metodo `public void Zapisi (string imeDatoteke)`, ki podatke o plavalcu zapiše v datoteko z imenom `imeDatoteke`. Vanjo naj se zapiše vrstica

```
Plavalec #ime #priimek plava #slog.
```

kjer namesto `#ime`, `#priimek` ter `#slog` vstavite prave vrednosti (v datoteki naj bo izključno ta vrstica).

Sestavite tudi testni program, kjer ustvarite plavalca `Danijel Ančimer`, ki najraje plava prsno, ter podatke o njem zapišite v datoteko `c:\plavalec.txt`.