PANEVROPSKI UNIVERZITET APEIRON
FAKULTET INFORMACIONIH TEHNOLOGIJA
BANJA LUKA

Seminarski rad

RAZNI ZADACI IZ C#

Nastavni predmet: ***Viši programski jezici i RAD alati***

Predmetni nastavnik:
Dražen Marinković

Student:
Siniša Božić
192-20/RITP

Banja Luka, 2021.

SADRŽAJ

UVOD

Ovaj seminarski rad obuhvata 20 primjera zadataka iz programskog jezika C#. Zadaci su preuzeti sa sajta Codewars, koji je jedan od najpopularnijih mjesta gdje korisnici širom svijeta unapređuju svoje programerske vještine rješavanjem brojnih problema u raznim programskim jezicima. Rješenja ovih zadataka su rezultat višemjesečnog autorovog rada, što se može utvrditi posjetom profila https://www.codewars.com/users/sbozich. Kako rješenja problema nisu vidljiva i nema načina da se vide sve dok korisnik uspješno ne riješi dati problem prolazeći sve predviđene testove, to je garancija da korisnik samostalno radi na rješenju zadataka, čime se najefikasnije istražuje i uči. Navedene zadatke i njima pripadajuća rješenja, kao i druge završene projekte sam postavio i na vlastiti GitHub profil koji se može naći na adresi https://github.com/sbozich/Codewars-Challenges.

Sami zadaci imaju svrhu razmišljanja u smislu konstruisanja algoritama i iako se radi o programima koji su konzolnog tipa, rješavanje navodi korisnika na istraživanje kako strukture podataka rade i kako im se može pristupiti i manipulisati u datom programskom jeziku. U tehničkom smislu radi se obično o kreiranju određene funkcije unutar zadatog okvira odnosno klase, to jest radi se o proceduralnom te funkcionalnom programiranju što se samih paradigmi tiče. Korisnik ima na raspolaganju predefinisane testove koje izvršava unutar browsera u programskom kompajleru. Po uspješnom kreiranju rješenja i njegovom testiranju, korisnik ima mogućnost da pregleda i ostala rješenja drugih, mnogobrojnih korisnika i da uporedi svoje sa ostalim rješenjima, što je još jedan vrlo koristan način učenja. Kodiranje se mora vježbati kucanjem koda i razmišljanjem o rješenjima te je u tom smislu korištenje ovakvih sajtova iz mog ugla vrlo korisno.

Za kraj, nisam navodio objašnjenja navedenih primjera jer bi to izašlo iz okvira seminarskog rada, već sam prezentovao vlastita rješenja navedenih problema kao takva.

ZADATAK 1[1]

Za zadani broj, utvrditi da li je to broj nastao kvadratnim stepenovanjem nekog broja, odnosno, da li za zadani broj postoji kvadratni korijen koji je cjelobrojan. Na primjer:

-1 => false

0 => true

3 => false

4 => true

25 => true

26 => false

Rješenje:

```
using System;

public class Kata
{
      public static bool IsSquare(int n)
      {
            double test=Math.Truncate((Math.Sqrt(n)));
            if (n<0)
            return false;
            else if (Math.Sqrt(n)==(int)(test))
            {
                  return true;
            }
            else
            return false;
      }
}
```

ZADATAK 2[2]

Implementirati funkciju koja oduzima jednu listu od druge i vraća rezultat. Funkcija treba da ukloni sve vrijednosti iz liste a, koje su prisutne u listi b, zadržavajući poredak unesenih podataka.

Na primjer:

(new int[] {1, 2}, new int[] {1}) => new int[] {2}

Ako je vrijednost prisutna u listi b, sva ponavljanja te vrijednosti moraju biti uklonjene iz liste a:

(new int[] {1, 2, 2, 2, 3}, new int[] {2}) => new int[] {1, 3}

1 Izvor: https://www.codewars.com/kata/54c27a33fb7da0db0100040e

2 Izvor: https://www.codewars.com/kata/523f5d21c841566fde000009

Rješenje:

```
using System.Collections.Generic;
using System.Linq;

public class Kata
{
      public static int[] ArrayDiff(int[] a, int[] b)
      {

            List<int> listA=new List<int>(a);
            List<int> listB=new List<int>(b);
            List<int> outcome = new List <int>();
            foreach(int x in listA)
            {
                  if(!listB.Contains(x))
                  {
                        outcome.Add(x);
                  }
            }

            int [] arr=outcome.ToArray();
            if (arr.Length==0)
            return new int[] {};
            else return arr;

      }
}
```

ZADATAK 3[3]

Implementirati funkciju koja će prekinuti tzv. "camel casing", tako što će na mjestu razdvajanja slova biti unesen razmak. Na primjer:

"camelCasing" => "camel Casing"

"identifier" => "identifier"

"" => ""

Rješenje:

```
public class Kata
{
      public static string BreakCamelCase(string str)
      {
            string outcome="";
            for (int i=0; i<str.Length;i++)
            {
                  if (char.IsUpper(str[i]))
                  {
                        outcome+=" ";
```

3 Izvor: https://www.codewars.com/kata/5208f99aee097e6552000148

```
                }
                outcome+=str[i];

            }
            return outcome.Trim();
      }
}
```

ZADATAK 4[4]

Za dati broj nivoa, kreirati ispis koji oponaša trokut. Karakteri su predstavljeni kao *. Na primjer, za dati broj 3 ispis bi bio slijedeći:

```
  *
 **
***
```

Rješenje:

```
using System.Collections.Generic;

public class Kata
{
      public static string[] TowerBuilder(int nFloors)
      {
            int blank=nFloors*2-1;
            List <string> lista=new List<string>();
            // create a list because of the array's indexing limitations
            int j;
            for(j=1;j<nFloors*2;j+=2) // looping from 1 with +2 to get an odd numbers
            {
                  //creating strings of spaces before stars and after and joining them with concantenation with stars
                  lista.Add((new string(' ', (blank-j)/2))+(new string('*', j)))+(new string(' ', (blank-j)/2)));
            }

            return lista.ToArray();
      }
}
```

ZADATAK 5[5]

Za dati argument koji je niz, kompletirati funkciju countSmileys koja vraća ukupan broj smajlija.

Pravila su slijedeća:

4 Izvor: https://www.codewars.com/kata/576757b1df89ecf5bd00073b
5 Izvor: https://www.codewars.com/kata/583203e6eb35d7980400002a

- svaki smajli mora imati par očiju, koji je prezentovan karakterima : ili ;. Smajli može imati nos ali i ne mora. Validni karakteri za nos su - ili ~. Svaki smajli mora imati usta koja trebaju biti označena sa) ili D.
- svi ostali krakteri nisu dozvoljeni, izuzev pomenutih.

Validni primjeri smajlija su: :) :D ;D :~), a neki smajliji koji nisu ispravni su ;(:> :} :]. Primjer funkcije:

countSmileys([':)', ';(', ';}', ':-D']); // treba vratiti 2;
countSmileys([';D', ':-(', ':-)', ';~)']); // treba vratiti 3;
countSmileys([';]', ':[', ';*', ':$', ';-D']); // treba vratiti 1;

Napomena: u slučaju praznog niza vratiti 0. Testovi neće obuhvatati input drugi od niza. Poredak karaktera u smajliju (oči, nos, usta) će uvijek biti isti.

Rješenje:

```
using System.Linq;

public static class Kata
{
      public static int CountSmileys(string[] smileys)
      {
            int count=0;
            if (smileys.Length==0) return count;
            else
            {
            string [] arrS={":)", ":D", ":-)", ":~)", ":-D", ":~D",
";)", ";D", ";-)", ";~)", ";-D", ";~D"};
                  for (int i=0; i<arrS.Length; i++)
                  count+=smileys.Count(s=>s ==arrS[i]);

                  return count ;
            }
      }
}
```

ZADATAK 6[6]

Redoslijed niza podataka treba obrnuti. Svaki segment je dug 8 bita. Na primjer:

11111111 00000000 00001111 10101010

6 Izvor: https://www.codewars.com/kata/569d488d61b812a0f7000015

(byte1) (byte2) (byte3) (byte4)

treba da postane

10101010 00001111 00000000 11111111
(byte4) (byte3) (byte2) (byte1)

Ukupan broj bita će uvijek biti umnoženik broja 8.
Podaci su dati u nizu kao na primjeru:
[1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,1,0,1,0,1,0]

Rješenje:

```
using System;
public class Kata
{
      public static int[] DataReverse(int[] data)
      {
            string dataString=string.Join("", data);
            string outcome="";
            for (int i=dataString.Length-8; i>=0; i-=8)
            outcome+=dataString.Substring(i,8);

            char [] charArr=outcome.ToCharArray();
            int [] result=new int[charArr.Length];
            for (int i=0; i<charArr.Length; i++)
            result[i]=Convert.ToInt32(charArr[i].ToString());

            return result;

      }
}
```

ZADATAK 7[7]

Napraviti funkciju koja uzima bilo koji pozitivni broj kao argument i vraća brojke u opadajućem redoslijedu. Odnosno, preurediti redoslijed kako bi se kreirao najveći mogući broj. Primjeri:

Input: 42145 Output: 54421

Input: 145263 Output: 654321

7 Izvor: https://www.codewars.com/kata/5467e4d82edf8bbf40000155

Input: 123456789 Output: 987654321

Rješenje:

```
using System;
using System.Collections.Generic;

public static class Kata
{
      public static int DescendingOrder(int num)
      {
            var digits = new List<int>();

            while (num > 0)
            {
                  digits.Add(num % 10); //use modulo to separate
digits
                  num /= 10;
            }

            digits.Sort(); // first sort from 0-9
            digits.Reverse(); // then sort from 9-0 which is asked in
this task

            int [] array=digits.ToArray();

            int finalScore = 0;
            for (int i = 0; i < array.Length; i++)
            {
                  finalScore          +=          array[i]          *
Convert.ToInt32(Math.Pow(10, array.Length-i-1));
            }


            return finalScore;
      }
}
```

ZADATAK 8[8]

Pangram je rečenica koja sadrži svaki karakter alfabeta barem jednom. Na primjer, rečenica: "The quick brown fox jumps over the lazy dog" je na engleskom jeziku pangram, jer koristi sva slova A-Z barem jednom (ne uzimajući u obzir razliku između velikih i malih slova).
Zadatak je za dati string detektovati da li je u pitanju pangram ili ne. Ignorisati brojeve i punktuacijske znake.

Rješenje:

8 Izvor: https://www.codewars.com/kata/545cedaa9943f7fe7b000048

```
using System;
using System.Collections.Generic;


public static class Kata
{
      public static bool IsPangram(string str)
      {
            str=str.ToLower(); //to avoid capital leters checking
            string alphabet="abcdefghijklmnopqrstuvwxyz";
            List<char>lista=new List<char>();
            for (int i=0; i<alphabet.Length;i++)
            {
                  if (str.Contains(alphabet[i]))
                  {
                        lista.Add(alphabet[i]);
                  }
            }
            bool outcome=false;
            if (lista.Count==26) outcome=true;
            return outcome;

      }
}
```

ZADATAK 9[9]

Cilj ovog zadatka je konvertovati string u novi string gdje je svaki karakter u novom stringu "(", ako se taj karakter pojavljuje samo jednom u prvom stringu, ili ")", ako se pojavljuje više od jednom u prvom stringu. Ignorisati kapitalizaciju. Primjeri:

"din" => "((("

"recede" => "()()()"

"Success" => ")())())"

"((@" => "))(("

Rješenje:

```
using System;
using System.Linq;
using System.Collections.Generic;

public class Kata
{
      public static string DuplicateEncode(string word)
      {

            List <string> lista=new List<string>();
```

9 Izvor: https://www.codewars.com/kata/54b42f9314d9229fd6000d9c

```
            List <string> lista2=new List<string>();
            List <int> lista3=new List<int>();
            word=word.ToLower();

            for(int i=0; i<word.Length;i++)
            lista2.Add(word[i].ToString());

            string [] wordArr=lista2.ToArray();

            List<string> listWord=wordArr.ToList();

            var distinctArray = lista2.Distinct().ToArray();

            if (word.Length==distinctArray.Length)
            for (int i=0; i<word.Length; i++)
            for(int j=0; j<i; j++)
            if (word[i]!=word[j])
            return new string('(', word.Length);


            // Find repeating chars.
            if (word.Length!=distinctArray.Length)
            for (int i=0; i<word.Length; i++)
            for (int j=0; j<i; j++)
            if (word[i]==word[j])
            lista.Add(word[i].ToString());

            var distinctArrayReps = lista.Distinct().ToArray();

            // Find the positions of the repeating chars.
            for (int i=0; i<distinctArrayReps.Length; i++)
            for (int j=0; j<word.Length; j++)
            if (distinctArrayReps[i]==word[j].ToString())
            lista3.Add(j);

            // Convert brackets to ASCII code to not interfere with
input with brackets.
            int leftbracketASCII=Convert.ToInt32('(');
            int rightbracketASCII=Convert.ToInt32(')');

            // Convert repeating chars with corresponding bracket
detected by ASCII code.
            // If we detect bracket only as a char, it will mess up
brackets count in a test string.
            for (int i=0; i<lista3.Count; i++)


listWord[lista3[i]]=Convert.ToInt32(char.Parse(")")).ToString();


            // Convert chars that do not repeat with corresponding
bracket.
            for (int i=0; i<listWord.Count; i++)
            if (listWord[i]!=rightbracketASCII.ToString())
            listWord[i]=leftbracketASCII.ToString();
```

```
            // Some needed conversions to get string output.
            List <string> outcLista=new List<string>();
            foreach (string x in listWord)
            outcLista.Add(((char)Convert.ToInt32(x)).ToString());
            string [] outcomeArr=outcLista.ToArray();

            string outcome = string.Join( "", outcLista.ToArray() );


            return outcome;
      }

}
```

ZADATAK 10[10]

Za dati brojčani niz, pronaći karakter koji se ponavlja neparni broj puta. Uvijek će biti jedna takva cifra. Primjeri:

[7] treba vratiti 7, jer se pojavljuje 1 put (neparni broj ponavljanja).

[0] treba vratiti 0, jer se pojavljuje 1 put (neparni broj ponavljanja).

[1,1,2] treba vratiti 2, jer se pojavljuje 1 put (neparni broj ponavljanja).

[0,1,0,1,0] treba vratiti 0, jer se pojavljuje 3 put (neparni broj ponavljanja).

[1,2,2,3,3,3,4,3,3,3,2,2,1] treba vratiti 4, jer se pojavljuje 1 put (neparni broj ponavljanja).

Rješenje:

```
using System;
using System.Linq;
using System.Collections.Generic;

namespace Solution
{
      class Kata
      {
            public static int find_it(int[] seq)
            {
                  Array.Sort(seq);

                  List<int> lista=new List<int>();

                  for (int i=0; i<seq.Length; i++)
                  {
                        // This adds in a list number of repetitions
of array seq's elements.
                        lista.Add(seq.Count(a => a == (seq[i])));
                  }
```

10 Izvor: https://www.codewars.com/kata/54da5a58ea159efa38000836

```
                // Loop finds first odd number in the list and puts
it at the last place.
                // According to the specifications, there is only
one number with an odd number of repetitions.
                //  This  finds  only  first  odd  number  which  is
repetition, not the actual number though!
                for (int i=0; i<lista.Count; i++)
                {
                     if (lista[i]%2!=0)
                     {
                          lista.Add(lista[i]);

                          break;

                     }

                }
                // Here we search for a position of a number in the
list, because the length of list and
                // original array seq is equal, we find the number.
                int    index    =    Array.IndexOf(lista.ToArray(),
lista.Last());

                return seq[index];



           }

      }
}
```

ZADATAK 11[11]

U ovom zadatku je zadan brojčani string, a cilj je pronaći najveći i najmanji broj. Primjeri:

Kata.HighAndLow("1 2 3 4 5"); // return "5 1"

Kata.HighAndLow("1 2 -3 4 5"); // return "5 -3"

Kata.HighAndLow("1 9 3 4 -5"); // return "9 -5"

Napomene:

Svi brojevi su tipa Int32, nema potrebe za provjerom. U ulaznom stringu će biti najmanje jedan broj. Izlazni string mora da sadrži dva broja sa praznim prostorom između njih, navodeći prvo najveći broj.

11 Izvor: https://www.codewars.com/kata/554b4ac871d6813a03000035

Rješenje:

```
using System;

public static class Kata
{
      public static string HighAndLow(string numbers)
      {
            string [] array=numbers.Split(" "); //convert to array
using spaces as delimiter
            int [] arrayInt=new int [array.Length];
            for (int i=0; i<array.Length;i++)
            {
                  arrayInt[i]=Convert.ToInt32(array[i]); //copy values
from string to int array with conv.
            }

            Array.Sort(arrayInt); //now we can easily sort the array
            string max=(arrayInt[arrayInt.Length-1]).ToString();
            string min=(arrayInt[0]).ToString();
            return (max+" "+min);
      }
}
```

ZADATAK 12[12]

Kreirati funkciju koja uzima tri argumenta, brojčane vrijednosti a,b,c. Funkcija treba da vrati true ako je sa ovim brojevima moguće sastaviti trougao, uzimajući svaku vrijednost kao dužinu stranice trougla, odnosno false, ako je sa ulaznim podacima to nemoguće.

Rješenje:

```
public class Triangle
{
      public static bool IsTriangle(int a, int b, int c)
      {
            if (  (a+b)>c && (a+c)>b && (b+c)>a  && a>0 && b>0 && c>0
)
            return true;
            else return false;
      }
}
```

ZADATAK 13[13]

Za dati ulazni string, konvertovati ga tako da svako početno slovo bude veliko. Primjer:

"How can mirrors be real if our eyes aren't real" => "How Can Mirrors Be Real If Our Eyes Aren't Real"

12 Izvor: https://www.codewars.com/kata/56606694ec01347ce800001b
13 Izvor: https://www.codewars.com/kata/5390bac347d09b7da40006f6

Rješenje:

```
using System;
using System.Globalization;

public static class JadenCase
{
      public static string ToJadenCase(this string phrase)
      {
            char [] array=phrase.ToCharArray();

            if (array.Length>=1)
            {
                  if (char.IsLower(array[0]))
                  {array[0]=char.ToUpper(array[0]);}
            }

            for (int i=1; i<array.Length; i++)
            {
                  if (array[i-1]==' ')
                  {
                        if (char.IsLower(array[i]))
                        {
                              array[i]=char.ToUpper(array[i]);
                        }
                  }
            }
            return (new string(array));
      }
}
```

ZADATAK 14[14]

Kreirati NxN matricu, veličine definisane u parametru funkcije. Na primjer, ako je veličina 3:

1 2 3

2 4 6

3 6 9

Povratna vrijednost je niz: [[1,2,3],[2,4,6],[3,6,9]]

Rješenje:

```
using System;

namespace Solution
{
      class Solution
```

14 Izvor: https://www.codewars.com/kata/534d2f5b5371ecf8d2000a08

```
	{
		public static int[,] MultiplicationTable(int size)
		{
			// Declaration of a 2D array.
			int [,] array=new int[size, size];

			// Loop from 1 to be able to multiple values later.
			for (int i=1; i<=size; i++)
			{
				for (int j=1; j<=size; j++ )
				{
					// Add -1 to start with index 0, since
loop begins from number 1.
					array[i-1, j-1]=i*j;
				}
			}

			return array;

		}
	}
}
```

ZADATAK 15[15]

Za dati broj, proračunati i vratiti broj puta za koji se navedeni broj može množenjem njegovih pojedinačnih cifara dovesti u jednocifren broj. Na primjer:

39 --> 3 (jer je 3*9 = 27, 2*7 = 14, 1*4 = 4)

999 --> 4 (jer je 9*9*9 = 729, 7*2*9 = 126, 1*2*6 = 12, i konačno 1*2 = 2)

4 --> 0 (jer je 4 već jednocifreni broj)

Rješenje 1:

```
using System;

public class Persist
{
	public static int Persistence(long n)
	{

		int product=1;
		int persistence=0;

		if (n<10) return persistence;
		else
		{
			while (n>9)
			{
				while (n!=0)
				{
```

15 Izvor: https://www.codewars.com/kata/55bf01e5a717a0d57e0000ec

```
                        product*= (int) n%10;
                        n=n/10;
                    }
                    persistence++;

                    n=product;
                    product=1;

                }

            }
            return persistence;

        }
}
```

Rješenje 2:

```
using System.Collections.Generic;

using System;

public class Persist
{
      public static int Persistence(long n)
      {

            if (n<10)
            {
                  return 0;
            }

            long persistence=1;
            while (n>0)
            {
                  persistence*=n%10;
                  n/=10;
            }
            // Using a recursive function.
            return 1+Persistence(persistence);

      }
}
```

ZADATAK 16[16]

Neki brojevi imaju interesantne karakteristike. Na primjer:

89 --> $8^1 + 9^2 = 89 * 1$

695 --> $6^2 + 9^3 + 5^4 = 1390 = 695 * 2$

46288 --> $4^3 + 6^4 + 2^5 + 8^6 + 8^7 = 2360688 = 46288 * 51$

16 Izvor: https://www.codewars.com/kata/5552101f47fc5178b1000050

Ako je dat prirodni broj n napisan kao abcd (cifre) i prirodni broj p, želimo naći prirodni broj k, koji ako postoji, je proizvod broja p i broja k. Drugim riječima, da li postoji prirodni broj k takav da važi:

(a ^ p + b ^ (p+1) + c ^(p+2) + d ^ (p+3) + ...) = n * k

Ako da, vratiti k, ako ne, vratiti -1.

Još primjera:

digPow(89, 1) treba vratiti 1 jer važi $8^1 + 9^2 = 89 = 89 * 1$

digPow(92, 1) treba vratiti -1 jer ne postoji k takvo da važi $9^1 + 2^2 = 92 * k$

digPow(695, 2) treba vratiti 2 jer važi $6^2 + 9^3 + 5^4 = 1390 = 695 * 2$

digPow(46288, 3) treba vratiti 51 jer važi $4^3 + 6^4 + 2^5 + 8^6 + 8^7 = 2360688 = 46288 * 51$

Rješenje:

```
using System;
using System.Collections.Generic;

public class DigPow {
     public static long digPow(int n, int p) {

          // Preserve original int n.
          int nn=n;
          List<int> lista=new List<int>() ;

          // Convert int n to list for easier manipulation.
          while (n != 0)
          {
               lista.Insert(0, n % 10);
               n = n / 10;
          }
          int [] array=lista.ToArray();
          int sum=0;

          // Sum digit powers in list.
          for (int i =0, j=p; i<array.Length; i++)
          {
               sum+=(int)Math.Pow(array[i],j);
               j++;
          }

          if (sum/nn*nn==sum)
          return sum/nn;
          else return -1;
     }
}
```

ZADATAK 17[17]

Bankomati obično dozvoljavaju 4- ili 6-cifreni PIN kod pri korištenju bankarskih kartica.

Funkcija treba da provjeri da li je ulazni string validan PIN kod u tom smislu. Na primjer:

"1234" --> true

"12345" --> false

"a234" --> false

Rješenje:

```
using System;
using System.Text.RegularExpressions;

public class Kata
{
      public static bool ValidatePin(string pin)
      {

            string pattern = "^\\d{4}$";
            string pattern2 = "^\\d{6}$";

            Regex rgx = new Regex(pattern);
            Regex rgx2 = new Regex(pattern2);

            if      ((pin.Length==4      ||      pin.Length==6)      &&
int.TryParse(pin.ToString(), out _))
            {
                  if ((rgx.IsMatch(pin) || (rgx2.IsMatch(pin))) )
                  {
                        return true;
                  }
                  else return false;
            }   else return false;

      }
}
```

ZADATAK 18[18]

Za dati brojčani niz sortirati neparne brojeve u rastućem redoslijedu dok parne brojeve ostaviti na datim pozicijama. Primjeri:

[7, 1] => [1, 7]
[5, 8, 6, 3, 4] => [3, 8, 6, 5, 4]
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0] => [1, 8, 3, 6, 5, 4, 7, 2, 9, 0]

17 Izvor: https://www.codewars.com/kata/55f8a9c06c018a0d6e000132
18 Izvor: https://www.codewars.com/kata/578aa45ee9fd15ff4600090d

Rješenje:

```
using System;
using System.Collections.Generic;

public class Kata
{
	public static int[] SortArray(int[] array)
	{
		List<int> listOdd=new List<int>();
		foreach (int i in array)
		if (i%2!=0)
		{
			listOdd.Add(i);
		}
		int [] oddArr=listOdd.ToArray();
		Array.Sort(oddArr);

		for (int i=0, j=0; i<array.Length; i++)
		{
			if (array[i]%2!=0)
			{
				array[i]=oddArr[j];
				j++;
			}
		}

		foreach(int i in array) Console.WriteLine(i);
		return array;
	}
}
```

ZADATAK 19[19]

Ako prvi argument završava sa drugim, vratiti true, inače false. Primjer:

solution('abc', 'bc') // returns true

solution('abc', 'd') // returns false

Rješenje:

```
public class Kata
{
	public static bool Solution(string str, string ending)
	{
		int lengthStr=str.Length-1;
		int lengthEnding=ending.Length-1;
		bool outcome;

		if (ending.Length==0)
		return true;
```

19 Izvor: https://www.codewars.com/kata/51f2d1cafc9c0f745c00037d

```
            else          if          (str.Contains(ending)          &&
(ending[lengthEnding].Equals(str[lengthStr])))
            outcome=true;
            else
            outcome=false;

            return outcome;
      }
}
```

ZADATAK 20[20]

Za dati brojčani niz, kreirati funkciju koja vraća sumu dva najmanja pozitivna broja. Na primjer:

[19, 5, 42, 2, 77] => 7.

[10, 343445353, 3453445, 3453545353453] => 3453455.

Rješenje:

```
using System;

public static class Kata
{
      public static int sumTwoSmallestNumbers(int[] numbers)
      {
            Array.Sort(numbers);

            return numbers[0]+numbers[1];
      }
}
```

20 Izvor: https://www.codewars.com/kata/558fc85d8fd1938afb000014

ZAKLJUČAK

U prezentovanim zadacima je vidljivo da su zahtjevi pred korisnikom manipulisanje osnovnim strukturama podataka kao što su stringovi, nizovi i liste. Iako se možda ovakvi zadaci čine jednostavnim, u smislu da ne sadrže konekcije ka bazama podataka i trećim izvorima te nisu grafički/GUI-orjentisane, njihovo rješavanje podrazumijeva rješavanje matematičko-programerskih problema i kreiranje složenih algoritama, ili, detaljnije poznavanje programskog jezika i njegovih ugrađenih mogućnosti (kao što je to klasa Linq u slučaju C#) koji na jednostavniji način omogućavaju rješavanje ovakvih problema. Nekada je bilo potrebno za pojedine probleme i višednevno razmišljanje, istraživanje mogućih solucija i njihova implementacija, a kao što je poznato, kroz pokušaje i greške korisnik puno više nauči nego kroz pasivno posmatranje video kurseva, čitanje programerskih knjiga i sl - što ne znači da ti resursi nisu korisni, već da je potrebno kombinovati više izvora znanja.

LITERATURA

[1] Murach's C#, Anne Boehm, Mike Murrach & Asoc. Inc., 2016

[2] SoloLearn C# Course, https://www.sololearn.com/learning/1080

[3] The C# Programming Yellow Book, Rob Miles, Independently published, 2018