# Решения на задачите от глава 9 на книгата "Въведение в програмирането със C#"

## Авторски колектив

**Иван Стоянов – <mad\_van@abv.bg>**

**Krasin Georgiev - <gkrasin@gmail.com>**

Angel Venchev - <angelvenchev@yahoo.com>

Tsvetomir Kirchev - <tsvetomir.kirchev@gmail.com>

Stefan Stoyanov - <stefan.asenov\_@abv.bg>

|  |  |
| --- | --- |
| 01. SayHello | |
| **Условие**  Напишете метод, който при подадено име отпечатва в конзолата "Hello, <name>!" (например "Hello, Peter!"). Напишете програма, която тества този метод дали работи правилно. | |
| **Описание на входа**  Въвеждаме кое да е име. | |
| **Описание на изхода**  Програмата изписва на екрана "Hello, <name>!" | |
| **Анализ на задачата**  Имаме метод, който въвежда името и след това го извикваме на екрана. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_10\_Methods  {  class Program  {  static void Main(string[] args)  {  string name;  name = InputName();  Console.WriteLine("Hello, {0}!", name);  }  private static string InputName()  {  string name;  Console.Write("Input your name ");  name = Console.ReadLine();  return name;  }  }  } | |
| **Тестове**  Въвеждаме кое да е име. | |
| **Вход** | **Изход** |
| Ivan  Ivan Stoyanov  Иван | Hello, Ivan!  Hello, Ivan Stoyanov!  Hello, Иван! |
|  | |
| 02. GetMax | |
| **Условие**  Създайте метод GetMax() с два целочислени (int) параметъра, който връща по-голямото от двете числа. Напишете програма, която прочита три цели числа от конзолата и отпечатва най-голямото от тях, изпол­звайки метода GetMax(). | |
| **Описание на входа**  Въвеждаме три цели числа подред. | |
| **Описание на изхода**  На конзолата се изписва най-голямото число | |
| **Анализ на задачата**  Сравняваме двете числа и по-голямото даваме като аргумент на същата функция с третото число. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_02.GetMax  {  class Program  {  static void Main(string[] args)  {  int a = int.Parse(Console.ReadLine());  int b = int.Parse(Console.ReadLine());  int c = int.Parse(Console.ReadLine());  Console.WriteLine(GetMax(GetMax(a,b), c));  }  private static int GetMax(int a, int b)  {  if (a > b) return a;  else return b;  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
| 1  2  3  4  2  6  -1  5  0 | 3  6  5 |
|  | |

|  |  |
| --- | --- |
| 03. LastDigitInWord | |
| **Условие**  Напишете метод, който връща английското наименование на послед­ната цифра от дадено число. Примери: за числото 512 отпечатва "two"; за числото 1024 – "four". | |
| **Описание на входа**  Въвеждаме цяло число. | |
| **Описание на изхода**  Изписва се последната цифра на въведеното число. | |
| **Анализ на задачата**  С функцията %10 взимаме последната цифра на дадено число и я проеверяваме таблично на коя дума отговаря. Ако числото е отрицателно взимаме обратната му стойност. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_03.LastDigitInWord  {  class Program  {  static void Main(string[] args)  {  int num = int.Parse(Console.ReadLine());  string lastDigit = GetLast(num % 10);  Console.WriteLine(lastDigit);  }  static string GetLast(int num)  {  if (num < 0) num = -num;  if (num == 0) return "Zero";  else if (num == 1) return "One";  else if (num == 2) return "Two";  else if (num == 3) return "Three";  else if (num == 4) return "Four";  else if (num == 5) return "Five";  else if (num == 6) return "Six";  else if (num == 7) return "Seven";  else if (num == 8) return "Eight";  return "Nine";  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
| 4  0  10  123456  -10  2 | Four  Zero  Zero  Six  Zero  Two |
|  | |
| 04. CountNumInArray | |
| **Условие**  Напишете метод, който намира колко пъти дадено число се среща в даден масив. Напишете програма, която проверява дали този метод работи правилно. | |
| **Описание на входа**  Първо въвеждаме колко числа искаме да въведем. След това въвеждаме числата и накрая въвеждаме търсеното число. | |
| **Описание на изхода**  На екрана се изписва колко пъти е срещнато търсеното число. | |
| **Анализ на задачата**  Въвеждаме колко числа искаме да въведем ако броят им е по-малък то 1 то прекъсваме изплънението. След това въвеждаме числата. И накрая въвеждаме търсеното число. Програмата минава по целия масив и накрая печата колко пъти е срещнато даденото число. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace CountNumInArray  {  class Program  {  static void Main(string[] args)  {  Console.Write("How many numbers you want to input - ");  int arrayLenght = int.Parse(Console.ReadLine());  if(arrayLenght<1) throw new Exception("You must enter some numbers!");  Console.WriteLine("Input the Numbers:");  int[] array = new int[arrayLenght];  for (int i = 0; i < array.Length; i++)  {  array[i] = int.Parse(Console.ReadLine());  }  Console.Write("Number to search for - ");  int numToSearch = int.Parse(Console.ReadLine());  Console.WriteLine("Found {0} times", CheckNum(array, numToSearch));  }  static int CheckNum(int[] array, int num)  {  int cnt = 0;  for (int i = 0; i < array.Length; i++)  {  if (array[i] == num) cnt++;  }  return cnt;  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
| How many numbers you want to input - 3  Input the Numbers:  1  2  3  Number to search for - 2  How many numbers you want to input - 0 | Found 1 times  Unhandled Exception: System.Exception: You must enter some numbers! |
|  | |

|  |  |
| --- | --- |
| 05. CheckTheNeiboursInArray | |
| **Условие**  Напишете метод, който проверява дали елемент, намиращ се на дадена позиция от масив, е по-голям, или съответно по-малък от двата му съседа. Тествайте метода дали работи коректно. | |
| **Описание на входа**  Първо въввеждаме колко числа искаме да въведем. След това въвеждаме числата и накрая въвеждаме желания индекс( от 0 ). | |
| **Описание на изхода**  Изписва се дали елемента на съответната позиция е по-голям, по-малък или нито едното от двете. | |
| **Анализ на задачата**  Въвеждаме броя на числата, които искаме да въведем. Ако е по-малък от 1 спираме изпълнението. След това въвеждаме числата и накрая въвеждаме индекса на елемента който желаем да проверим. Ако е индекса е 0 или последния възможен казваме че това е първият елемент или последният и той няма два съседа. Ако индекса е по-голям от дължината на масива казваме, че няма такава позиция. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_05.CheckTheNeiboursInArray  {  class Program  {  static void Main(string[] args)  {  Console.Write("How many numbers you want to input - ");  int arrayLenght = int.Parse(Console.ReadLine());  if (arrayLenght < 1) throw new Exception("You must enter some numbers!");  Console.WriteLine("Input the Numbers:");  int[] array = new int[arrayLenght];  for (int i = 0; i < array.Length; i++)  {  array[i] = int.Parse(Console.ReadLine());  }  Console.Write("Input the positon to check - ");  int position = int.Parse(Console.ReadLine());  if (position < 0 || position > arrayLenght) throw new Exception("Wrong position!");  CheckNeighbors(array, position);  }  static void CheckNeighbors(int[] array, int position)  {  if (position == 0)  {  Console.WriteLine("this is first element");  return;  }  if (position == array.Length-1)  {  Console.WriteLine("this is last element");  return;  }  if (array[position] > array[position - 1] && array[position] > array[position + 1])  {  Console.WriteLine("bigger then neibours");  return;  }  if (array[position] < array[position - 1] && array[position] < array[position + 1])  {  Console.WriteLine("smaller then neibors");  return;  }  else Console.WriteLine("not bigger or smaller then neibours"); return;  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
| How many numbers you want to input - 1  Input the Numbers:  1  Input the positon to check – 0  How many numbers you want to input - 2  Input the Numbers:  1  2  Input the positon to check – 1  How many numbers you want to input - 3  Input the Numbers:  1  2  3  Input the positon to check – 1  How many numbers you want to input - 3  Input the Numbers:  1  2  1  Input the positon to check – 1  How many numbers you want to input - 3  Input the Numbers:  1  2  3  Input the positon to check - 4 | this is first element  this is last element  not bigger or smaller then neibours  bigger then neibours  Unhandled Exception: System.Exception: Wrong position! |
|  | |
| 06. FirstElementInArrayBiggerThenNeibours | |
| **Условие**  Напишете метод, който връща позицията на първия елемент на масив, който е по-голям от двата свои съседи едновременно, или -1, ако няма такъв елемент. | |
| **Описание на входа**  Въвеждаме колко числа искаме да въведем. След това въвеждаме числата. | |
| **Описание на изхода**  На екрана се изписва позицията на елемента, който отговаря на условието или съответно -1. | |
| **Анализ на задачата**  След като въведем числата започваме да ги обхождаме от второто до предполседното(само тези числа имат по два съседа). В момента, в който намерим такъв елемент спираме обхождането и изпечатваме позицията. Ако не намерим такъв елемент до края, изпечатваме -1. (Използваме задача 5) | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_06\_Use05  {  class Program  {  static void Main(string[] args)  {  Console.Write("How many numbers you want to input - ");  int arrayLenght = int.Parse(Console.ReadLine());  if (arrayLenght < 1) throw new Exception("You must enter some numbers!");  Console.WriteLine("Input the Numbers:");  int[] array = new int[arrayLenght];  for (int i = 0; i < array.Length; i++)  {  array[i] = int.Parse(Console.ReadLine());  }  int result=CheckAllNeighbors(array);  if(result!=-1) Console.WriteLine("Found at position - {0}",result);  else Console.WriteLine("Not found");  }  static int CheckAllNeighbors(int[] array)  {  for (int i = 1; i < array.Length-1; i++)  {  if (CheckNeighbors(array, i)) return i;  }  return -1;  }  static bool CheckNeighbors(int[] array, int position)  {  if (array[position] > array[position - 1] && array[position] > array[position + 1])  {  return true;  }  else return false;  }  }  } | |
| **Тестове**  Интересни са няколко случая например ако дължината на масива е 1 или 2. | |
| **Вход** | **Изход** |
| How many numbers you want to input - 3  Input the Numbers:  1  2  3  How many numbers you want to input - 4  Input the Numbers:  1  2  2  1  How many numbers you want to input - 5  Input the Numbers:  1  2  3  2  1  How many numbers you want to input - 2  Input the Numbers:  1  2  How many numbers you want to input - 1  Input the Numbers:  1 | Not found  Not found  Found at position – 2  Not found  Not found |
|  | |

|  |  |
| --- | --- |
| 07. ReverseNumber | |
| **Условие**  Напишете метод, който отпечатва цифрите на дадено десетично число в обратен ред. Например 256, трябва да бъде отпечатано като 652. | |
| **Описание на входа**  Въвеждаме int число. | |
| **Описание на изхода**  На екрана се изписва обърнатото число. | |
| **Анализ на задачата**  Въвеждаме едно число и бройм неговите цифри. След това правим един цикъл в който закачаме числата отзад напред имножени по 10 на съотвената степен. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_07.ReverseNumber  {  class Program  {  static void Main(string[] args)  {  int number = int.Parse(Console.ReadLine());  int reverse = Reverse(number);  Console.WriteLine(reverse);  }  static int CountDigits(int num)  {  int cnt = 0;  while (num != 0)  {  num /= 10;  cnt++;  }  return cnt;  }  static int Reverse(int num)  {  int cnt = CountDigits(num);  int tmpNum = 0;  for (int i = cnt-1; i >=0 ; i--)  {  tmpNum += (num % 10) \* (int)Math.Pow(10, i);  num /= 10;  }  return tmpNum;  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
| 123  12345  -1234  1234213  0  1 | 321  54321  -4321  3124321  0  1 |
|  | |
| 08. SumTwoBigIntegers | |
| **Условие**  Напишете метод, който пресмята сумата на две цели положителни цели числа. Числата са представени като масив от цифрите си, като последната цифра (единиците) са записани в масива под индекс 0. Направете така, че метода да работи за числа с дължина до 10 000 цифри. | |
| **Описание на входа**  Въвеждаме два произволно дълги стринга от цифри от 1до9 | |
| **Описание на изхода**  На екрана се изписва сбора на двете числа. | |
| **Анализ на задачата**  Пазим двете числа в масив от char. При извикване на финкцията за събиране се заделя нова памет от масив от char[] като дължината му е дължината на по дългото от двете числа +1. В новия масив събираме числата на съответните позиции и ако сбора е по гилям от 9 си пазим едно на ум за следваштото събиране. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_08.BigInteger  {  class Program  {  static void Main(string[] args)  {  string num1 = Console.ReadLine();  string num2 = Console.ReadLine();  int[] arrayNum1 = new int[num1.Length];  int[] arrayNum2 = new int[num2.Length];  ToArray(num1, arrayNum1);  ToArray(num2, arrayNum2);  int[] sum = new int[Math.Max(num1.Length, num2.Length) + 1];  if (arrayNum1.Length > arrayNum2.Length)  {  Array.Copy(arrayNum2, sum, arrayNum2.Length);  SumArrays(sum, arrayNum1);  }  else  {  Array.Copy(arrayNum1, sum, arrayNum1.Length);  SumArrays(sum, arrayNum2);  }  Array.Reverse(sum);  int i = 0;  if (sum[i] == 0) i++;  for (; i < sum.Length; i++)  {  Console.Write(sum[i]);  }  }  private static void SumArrays(int[] sum, int[] array1)  {  int oneOnMind = 0;  for (int i = 0; i < array1.Length; i++)  {  if ((sum[i] + array1[i] + oneOnMind) > 9)  {  sum[i] = ((sum[i] + array1[i] + oneOnMind) % 10);  oneOnMind = 1;  }  else  {  sum[i] = ((sum[i] + array1[i] + oneOnMind) % 10);  oneOnMind = 0;  }  }  if (oneOnMind == 1) sum[sum.Length - 1] = 1;  }  private static void ToArray(string num1, int[] arrayNum1)  {  for (int i = 0, j = num1.Length - 1; j >= 0; i++, j--)  {  arrayNum1[i] = num1[j] - '0';  }  }  }  } | |
| **Тестове**  Числата са произволно дълги, но винаги трябва да са положителни. | |
| **Вход** | **Изход** |
| 12345  12345  1234567890  12  987654321  123456789  0  0  1  1  99999999999999999999999999999999999  1 | 24690  1234567902  1111111110  0  2  100000000000000000000000000000000000 |
|  | |

|  |  |
| --- | --- |
| 09. findMaxInArrayAndSort | |
| **Условие**  Напишете метод, който намира най-големия елемент в част от масив. Използвайте метода за да сортирате възходящо/низходящо даден масив. | |
| **Описание на входа**  На първия ред въвеждаме колко числа искаме да въведем и на всеки следващ ред въвеждаме по едно от числата. На последния ред въвеждаме позицията след която искаме да търсим. | |
| **Описание на изхода**  На първия ред се изписва най-голямото намерено число в масива от дадена позиция нататък. | |
| **Анализ на задачата**  Имаме един цикък и в него от дадена позиция нататък започчваме да проверяваме елементите един по един. Ако съответния елемент е по голям го пазим до края на масива и накрая извеждаме неговата стойност. За да сортираме масива с тази функция това което трябва да направим е като намерим най-големия елемент да го разменим с първия. След това търсим най-големи от втория нататък и отново го разменяма като накрая се получава сортиран масив низходящо. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace BinToHex  {  class Program  {  static void Main(string[] args)  {  string binNum1 = Console.ReadLine();  char[] binNum = binNum1.ToCharArray();  Array.Reverse(binNum);  if (binNum1.Length < 4 && int.Parse(binNum1) == 0) Console.WriteLine(0);  else  {  char[] hex = new char[(binNum.Length / 4) + 1];  hex = ToHex(binNum);  //print  int i = 0;  while (hex[i] == 0) i++;  if (hex[i] == '0') i++;  for (int j = i; j < hex.Length; j++)  {  Console.Write(hex[j]);  }  Console.WriteLine();  }  }  private static char[] ToHex(char[] binNum)  {  char[] tmpBig = new char[((binNum.Length / 4) + 1) \* 4];  short step = 0;  int length = ((binNum.Length / 4) + 1);  for (int i = 0; i < length; i++)  {  int little=0;  int num = 0;  for (int j = i\*4; j < (i\*4)+4; j++) | |
| **Тестове**  Числото винаги е положително. Тестваме с гранични случаи 0, 1, 10, 1111, 11111 | |
| **Вход** | **Изход** |
| 3  1  2  3  2  1  1  0  4  1  2  3  4  6 | 3  3 2 1  1  1  -1  4 3 2 1 |
|  | |
| 10. n! | |
| **Условие**  Напишете програма, която пресмята и отпечатва n! за всяко n в интервала [1…100]. | |
| **Описание на входа** | |
| **Описание на изхода**  На всеки нов ред се изписва факториела на 1,2,3...100 | |
| **Анализ на задачата**  Метода винаги започва от едно умножава по 2 след това по 3 и така докато не стигне до желаното число и след това всичко започва отначало. Резултата се пази в масив от 1000 елемента следователно можем да съберем число дълго 1000 цифри. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_10.n\_  {  class Program  {  static void Main(string[] args)  {  for (int num = 1; num < 100; num++)  {  int[] arrayNum = new int[1000];  arrayNum[0] = 1;  for (int i = 1; i <= num; i++)  {  Multiply(arrayNum, i);  }  Array.Reverse(arrayNum);  int j = 0;  while (arrayNum[j] == 0) j++;  for (; j < arrayNum.Length; j++)  {  Console.Write(arrayNum[j]);  }  Console.WriteLine();  }  }  private static void Multiply(int[] array, int a)  {  int oneOnMind = 0;  int i = 0;  for (; i < 1000; i++)  {  int temp = (array[i] \* a + oneOnMind);  if (temp > 9)  {  array[i] = (temp % 10);  oneOnMind = temp / 10;  }  else  {  array[i] = (temp % 10);  oneOnMind = 0;  }  }  if (oneOnMind != 0) array[i] = oneOnMind;  }    }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  | 1  2  6  24  120  720  5040  40320  362880  3628800  39916800  479001600  6227020800  87178291200  1307674368000  20922789888000  355687428096000  6402373705728000  121645100408832000  И т.н. |
|  | |

|  |  |  |  |
| --- | --- | --- | --- |
| 11. ProgramWithMenu | | | |
| **Условие**  Напишете програма, която решава следните задачи:   * Обръща последователността на цифрите на едно число. * Пресмята средното аритметично на дадена поредица от числа. * Решава линейното уравнение a \* x + b = 0.   Създайте подходящи методи за всяка една от задачите.  Напишете програмата така, че на потребителя да му бъде изведено текстово меню, от което да избира коя от задачите да решава.  Направете проверка на входните данни:   * Десетичното число трябва да е неотрицателно. * Редицата не трябва да е празна. * Коефициентът a не трябва да е 0. | | | |
| **Описание на входа** Въвеждаме число от 1 до 3 (зависи каква операция искаме)  Ако въведем 1 – въвеждаме едно число.  Ако въведем 2 – въвеждаме колко числа ще въведем и след това въвеждаме числата едно по едно  Ако въведем 3 – първо въвеждаме „а”, а след това ”в”. | | | |
| **Описание на изхода**  Ако сме въвели 1 – извежда се обърнатото число  Ако сме въвели 2 – извецда се средно аритметичното на редицата  Ако сме въвели 3 – се пресмята х | | | |
| **Анализ на задачата**  В зависимост от избора ни се извиква съответната функция (използват се предходните задачи) | | | |
| **Решение (сорс код)** | | | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_13.ProgramWithMenu  {  class Program  {  static void Main(string[] args)  {  Console.WriteLine("If you want to reverse a number press 1");  Console.WriteLine("If you want to calculata the avarege of sequence press 2");  Console.WriteLine("If you want to solves a linear equation a \* x + b = 0 press 3");  Console.Write("Choose option - ");  int option = int.Parse(Console.ReadLine());  while (option < 1 || option > 3)  {  Console.Write("Wrong option! Input again ");  option = int.Parse(Console.ReadLine());  }    if (option == 1)  {  Option1();  }  else if (option == 2)  {  Option2();  }  else  {  Option3();  }  }  private static void Option3()  {  Console.Write("Input number a ");  int a = int.Parse(Console.ReadLine());  while (a == 0)  {  Console.Write("Wrong! Input again ");  a = int.Parse(Console.ReadLine());  }  Console.Write("Input number b ");  int b = int.Parse(Console.ReadLine());  Console.WriteLine("x= {0}", (-b / (double)a));  }  private static void Option2()  {  Console.Write("Numbers to sum ");  int n = int.Parse(Console.ReadLine());  while (n < 1)  {  Console.Write("Wrong! Input again ");  n = int.Parse(Console.ReadLine());  }  int[] array = new int[n];  double sum = 0;  for (int i = 0; i < n; i++)  {  Console.Write("Input number {0} ", i + 1);  array[i] = int.Parse(Console.ReadLine());  sum += array[i];  }  //Console.WriteLine("Sum is " + sum);  Console.WriteLine("Avg is " + sum / n);  }  private static void Option1()  {  Console.Write("Write the number - ");  int num = int.Parse(Console.ReadLine());    Console.WriteLine("Old Number is " + num);  Console.WriteLine("New number is " + Reverse(num));  }  static int CountDigits(int num)  {  int cnt=0;  while (num != 0)  {  num /= 10;  cnt++;  }  return cnt;  }  static int Reverse(int num)  {  int cnt = CountDigits(num);  int tmpNum = 0;  for (int i = cnt-1; i >=0 ; i--)  {  tmpNum += (num % 10) \* (int)Math.Pow(10, i);  num /= 10;  }  return tmpNum;  }  }  } | | | |
| **Тестове** | | | |
| **Вход** | | | **Изход** |
| If you want to reverse a number press 1  If you want to calculata the avarege of sequence press 2  If you want to solves a linear equation a \* x + b = 0 press 3  Choose option - 4  Wrong option! Input again 1  Write the number – 123  Choose option - 2  Numbers to sum 3  Input number 1 2  Input number 2 5  Input number 3 7  Choose option - 3  Input number a 0  Wrong! Input again 1  Input number b 5 | | | Old Number is 123  New number is 321  Avg is 4.66666666666667  x= -5 |
|  | | | |
| 12. AddPolynom | | | |
| **Условие**  Напишете метод, който събира два полинома с цели коефициенти, например (3x2 + x - 3) + (x - 1) = (3x2 + 2x - 4). | | | |
| **Описание на входа**  Въвеждаме два полинома като поредица от техните коефициенти като започваме от най-малкия.  Например х2 + 2х + 3 се представя като 321 | | | |
| **Описание на изхода**  На екрана се извежда полученият полином. | | | |
| **Анализ на задачата**  Събираме получените полиномите позиция по позиция като нанрая ако единия е по дълъг от него просто си го преписваме и накрая извеждаме резултата. | | | |
| **Решение (сорс код)** | | | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_11.Unkown  {  class Program  {  static void PrintPolynom(int[] pol)  {  for (int i = pol.Length - 1; i > 0; i--)  {  if (pol[i] != 0)  Console.Write("{0}\*x^{1} + ", pol[i], i);  }  Console.WriteLine(pol[0]);  }  static void AddPolynom(string first, string second)  {  int resLen;  int indexOut;  string tmp;//points to largest array  if (first.Length >= second.Length)  {  resLen = first.Length;  indexOut = second.Length;  tmp = first;  }  else  {  resLen = second.Length;  indexOut = first.Length;  tmp = second;  }  int[] result = new int[resLen];  for (int i = 0; i < indexOut; i++)  {  result[i] = (first[i] - '0') + (second[i] - '0');  }  for (int j = indexOut; j < result.Length; j++)  {  result[j] = (tmp[j] - '0');  }  PrintPolynom(result);  }  static void Main(string[] args)  {  Console.WriteLine("Enter two polynomials as a sequance of their coefficiens from the lowest power(x^0).");  string firstPol = Console.ReadLine();  string secondPol = Console.ReadLine();  AddPolynom(firstPol, secondPol);  Console.WriteLine();  }  }  } | | | |
| **Тестове** | | | |
| **Вход** | **Изход** | | |
| 123  123  123  1234  1234567  1 | 6\*x^2 + 4\*x^1 + 2  4\*x^3 + 6\*x^2 + 4\*x^1 + 2  7\*x^6 + 6\*x^5 + 5\*x^4 + 4\*x^3 + 3\*x^2 + 2\*x^1 + 2 | | |
|  | | | |
| 13. MultPolynom | | | |
| **Условие**  Напишете метод, който умножава два полинома с цели коефициенти, например (3x2 + x - 3) \* (x - 1) = (3x3 - 2x2 - 4x + 3). | | | |
| **Описание на входа**  Въвеждаме два полинома като поредица от техните коефициенти като започваме от най-малкия.  Например х2 + 2х + 3 се представя като 321 | | | |
| **Описание на изхода**  На екрана се изписва полученият полином | | | |
| **Анализ на задачата**  Взимаме първото число на първия полином и го умножаваме с втория след това второта число и го умножаваме с втория и резултата събираме на едно място. | | | |
| **Решение (сорс код)** | | | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_12.Sum\_Sub\_Mul\_Arrays  {  class Program  {  static void PrintPolynom(int[] pol)  {  for (int i = pol.Length - 1; i > 0; i--)  {  if (pol[i] != 0)  Console.Write("{0}\*x^{1} + ", pol[i], i);  }  Console.WriteLine(pol[0]);  }  static void MultPolynom(string first, string second)  {  int[] mult = new int[(first.Length) \* (second.Length) + 1];  int pow = 0;  for (int i = 0; i < first.Length; i++)  {  for (int j = 0; j < second.Length; j++)  {  pow = i + j;  mult[pow] += (first[i] - '0') \* (second[j] - '0');  }  }  PrintPolynom(mult);  }  static void Main(string[] args)  {  Console.WriteLine("Enter two polynomials as a sequance of their coefficiens from the lowest power(x^0).");  string firstPol = Console.ReadLine();  string secondPol = Console.ReadLine();    MultPolynom(firstPol, secondPol);  Console.WriteLine();  }  }  } | | | |
| **Тестове** | | | |
| **Вход** | | **Изход** | |
| 123  123  123  0  12  1  101  12  101  101 | | 9\*x^4 + 12\*x^3 + 10\*x^2 + 4\*x^1 + 1  0  2\*x^1 + 1  2\*x^3 + 1\*x^2 + 2\*x^1 + 1  1\*x^4 + 2\*x^2 + 1 | |
|  | | | |