# Решения на задачите от глава 5 на книгата "Въведение в програмирането със C#"

Предлагаме ви решения на задачите от [книгата "Въведение в програ­мирането със C#"](http://www.introprogramming.info), заедно с анализ на задачата, описание на използва­ните идеи, алгоритми,подходи за решаване и тестове.

## Авторски колектив

**Георги Иванов Петров**

georgi.iv.petrov@gmail.com

**Георги Красимиров Георгиев**

georgievgeorgy@yahoo.com

**Мартин Борисов Василев**

martin\_bv@abv.bg

**Цветомир Йорданов Тодоров**

todorov.ceco@gmail.com

# Решение и описание на задачите

|  |  |
| --- | --- |
| Задача 1. Извеждане на по-голямото от две числа | |
| **Условие**  Да се напише if-конструкция, която проверява стойността на две целочислени променливи и разменя техните стойности, ако стойността на първата променлива е по-голяма от втората  Пример: **5 , 4** 🡪 4 , 5 | |
| **Описание на входа**  Входните данни се въвеждат на две реда, като на първия ред се въвежда първата променлива, а на втория ред се възвежда втората променлива. | |
| **Описание на изхода**  Изходът се извежда на два реда, където на първия ред се извежда по-малкото от двете числа, а на втория се извежда по-голямото. | |
| **Анализ на задачата**  С условната конструкция (**if**) се сравняват стойностите на променливите **firstNumber** и **secondNumber.** Aко **firstNumber** е по-голяма от **secondNumber** :   * Помощната променливата **comparator** приема стойността на **firstNumber**. * Променливата **firstNumber** приема стойността на **secondNumber**. * Променливата **secondNumber** приема стойността на **comparator**.   Така двете основни промеливи си разменят стойносттите.  Ако **secondNumber** е по-голяма от **firstNumber**, кодът в условната конструкция не се изпълнява.  Накрая се отпечатват стойносттите на **firstNumber** и **secondNumber.** | |
| **Решение (сорс код)** | |
| using System;  class ExchangeNumbers  {  static void Main()  {  // Read the input numbers  Console.WriteLine("Please input the first integer, a = ?");  long firstNumber = long.Parse(Console.In.ReadLine());    Console.WriteLine("Please input the second integer, b = ?");  long secondNumber = long.Parse(Console.In.ReadLine());  // Check if the first number is bigger, than the second number  if (firstNumber > secondNumber)  {  long comparator = firstNumber;  firstNumber = secondNumber;  secondNumber = comparator;  }  // Print the output  Console.WriteLine("a: " + firstNumber + "\nb: " + secondNumber);  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Примерният вход от условието на задачата * Първата променлива да е по – малка от втората – тогава променливите, трябва да запазят стойносттите си. * Равни стойностти на променливите – тогава променливите, трябва да запазят стойносттите си. * Празни стойностти – резултатът са празни стойностти | |
| **Вход** | **Изход** |
| 5  4 | 4  5 |
| **Вход** | **Изход** |
| -100  200 | -100  200 |
| **Вход** | **Изход** |
| 4000000  4000000 | 4000000  4000000 |
| **Вход** | **Изход** |
| (празни стойностти) | (празни стойностти) |

|  |  |
| --- | --- |
| Задача 2. Извеждане на знака от произведението на 3 реални числа | |
| **Условие**  Напишете програма, която показва знака (+ или -) от произведението на три реални числа, без да го пресмята. Използвайте последователност от if оператори. | |
| **Описание на входа**  Въвеждат се 3 числа, всяко на нов ред. | |
| **Описание на изхода**  Ако произведението е негативно, на козолата се изкарва:  The product of the 3 numbers is negative  Ако произведението е позитивно, на козолата се изкарва:  The product of the 3 numbers is positive  Ако произведението е 0, на козолата се изкарва:  0 | |
| **Анализ на задачата**  Въвеждат се 3 числа и се запазват в променливите **a**, **b**, **c**. С оператора if се проверява дали и трите числа са равни на 0. Ако това е така, на конзолата се отпечатва 0.  В противен случай се прави следната проверка:   * Дали едно от числата е отрицателно, а другите две положителни * Дали и трите числа са отирцателни   Ако едното от горните две условия е изпълнено, се отпечатва, че числото е отрицателно. В противен случай се отпечатва, че числото е положително. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class PlusOrMinus  {  static void Main(string[] args)  {  //on 3 seperate input lines are the 3 integers  int a = int.Parse(Console.ReadLine());  int b = int.Parse(Console.ReadLine());  int c = int.Parse(Console.ReadLine());  //on the output line should be "+" if the product is positive, "-" if negative and, "0" if zero  if (a == 0 || b == 0 || c == 0)  {  Console.WriteLine(0);  }  else if ((a < 0 && b > 0 && c > 0) || (b < 0 && a > 0 && c > 0) ||  (c < 0 && b > 0 && a > 0) || (a < 0 && b < 0 && c < 0))  {  Console.WriteLine("The product of the 3 numbers is negative");  }  else Console.WriteLine("The product of the 3 numbers is positive");  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * 0 0 0 ->0 * 5 4 -> The product of the 3 numbers is positive * -1 –45 -2 -> The product of the 3 numbers is negative * + 1 – 3 + 1 -> The product of the 3 numbers is negative | |
| **Вход** | **Изход** |
| 0  0 | 0 |
| 5  4 | The product of the 3 numbers is positive |
| **Вход** | **Изход** |
| -1  -45  -2 | The product of the 3 numbers is negative |
| **Вход** | **Изход** |
| 1  -3  1 | The product of the 3 numbers is negative |

|  |  |
| --- | --- |
| Задача 3. Извеждане на най-голямото от три числа | |
| **Условие**  Напишете програма, която намира най – голямото по стойност число, измежду три дадени числа  Пример: 5 , 4 , **70** 🡪 **70** | |
| **Описание на входа**  Входните данни се въвеждат на три реда, като на първия ред се въвежда първата променлива, на на втория ред се възвежда втората променлива, а на третия ред се възвежда третата променлива. | |
| **Описание на изхода**  Изходът се извежда на един единствен ред, където се извежда най- голямото от трите числа. | |
| **Анализ на задачата**  С условната конструкция (**if**) се сравняват стойностите на променливите **firstNumber, secondNumber** и **thirdNumber**  и дали **firstNumber** е най-голяма, измежду тях:   * Ако е най-голяма, се извежда стойността на **firstNumber**.   Ако не е, с условната конструкция (**else if**) се сравняват стойностите на променливите **firstNumber, secondNumber** и **thirdNumber**  и дали **secondNumber** е най-голяма, измежду тях:   * Ако е най-голяма, се извежда стойността на **secondNumber**.   Ако и двата по – горни случая не са верни то, с условната конструкция (**else**) се прихваща и третият случай, в който третата променлива е най - голяма.   * Извежда стойността на **thirdNumber.**   Ако две или повече променливи имат еднаква стойност и са най-голямото число, се извежда първата зададена и най-голяма променлива. | |
| **Решение (сорс код)** | |
| using System;  {  static void Main()  {  // Read the input  Console.WriteLine input ("Please input the first number, a = ?");  decimal firstNumber = decimal.Parse(Console.In.ReadLine());  Console.WriteLine("Please input the second number, b = ?");  decimal secondNumber = decimal.Parse(Console.In.ReadLine());  Console.WriteLine("Please input the third number, c = ?");  decimal thirdNumber = decimal.Parse(Console.In.ReadLine());  // Check the Biggest number  if (firstNumber >= secondNumber && firstNumber >= thirdNumber)  {  Console.WriteLine("the biggest number is the first number -> {0}", firstNumber);  }  else if (secondNumber >= thirdNumber && secondNumber >= firstNumber)  {  Console.WriteLine("the biggest number is the second number -> {0}", secondNumber);  }  else  {  Console.WriteLine("the biggest number is the third number -> {0}", thirdNumber);  }    }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Примерният вход от условието на задачата * Две равни, най - големи стойностти – тогава се извежда първата зададена такава. * Три равни, най - големи стойностти – тогава се извежда първата зададена такава. * Празни стойностти – резултатът са празни стойностти | |
| **Вход** | **Изход** |
| 5  4  **70** | 70 |
| **Вход** | **Изход** |
| 9  **505**  505 | 505 |
| **Вход** | **Изход** |
| **1234567**  1234567  1234567 | 1234567 |
| **Вход** | **Изход** |
| (празни стойностти) | (празни стойностти) |

|  |  |
| --- | --- |
| Задача 4. Извеждане на най-голямото от три числа | |
| **Условие**  Сортирайте 3 реални числа в намаляващ ред. Използвайте вложени if оператори. | |
| **Описание на входа**  Въвеждат се три числа, всяко на нов ред. | |
| **Описание на изхода**  Въведените числа се отпечатват всяко на нов ред и сортирани от най-голямо към най-малко. | |
| **Анализ на задачата**  Последователно се сравнява всяка тройка числа, като се търси кое от тях е по-голямо от другите две. Щом то бъде намерено, се проверяват другите в числа (по-малки от него). В зависимост от това което е по-голямо от тях, конкретната последователност вече е ясна и се изписва на екрана. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class SortingThreeNumbersInDescendingOrder  {  static void Main(string[] args)  {  // on the 3 input lines are the 3 numbers  double a = double.Parse(Console.ReadLine());  double b = double.Parse(Console.ReadLine());  double c = double.Parse(Console.ReadLine());  //first checking for the biggest doubleeger(or equal) and than checking the bigger from the last 2 with nested ifs  // on the 3 output lines are the sorted doubleegers  if ((a >= b) && (a >= c))  {  if (b >= c)  {  Console.WriteLine(a);  Console.WriteLine(b);  Console.WriteLine(c);  }  else  {  Console.WriteLine(a);  Console.WriteLine(c);  Console.WriteLine(b);  }  }  else if ((b >= a) && (b >= c))  {  if (a >= c)  {  Console.WriteLine(b);  Console.WriteLine(a);  Console.WriteLine(c);  }  else  {  Console.WriteLine(b);  Console.WriteLine(c);  Console.WriteLine(a);  }  }  else if ((c >= b) && (c >= a))  {  if (b >= a)  {  Console.WriteLine(c);  Console.WriteLine(b);  Console.WriteLine(a);  }  else  {  Console.WriteLine(c);  Console.WriteLine(a);  Console.WriteLine(b);  }  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * 1 5 3 -> 5 3 1 * 0 4 0 -> 4 0 0 * - 53 -98 -4 -> -4 -53 -98 | |
| **Вход** | **Изход** |
| 1  5  3 | 5  3  1 |
| **Вход** | **Изход** |
| 0  4  0 | 4  0  0 |
| **Вход** | **Изход** |
| -53  -98  -4 | -4  -53  -98 |

|  |  |
| --- | --- |
| Задача 5. Извеждане на името на цифра | |
| **Условие**  Напишете програма, която за дадена цифра (0-9), зададена като вход, извежда името на цифрата на български език.  Пример: 5 🡪 Пет | |
| **Описание на входа**  Входните данни се въвеждат на един единствен ред. | |
| **Описание на изхода**  Изходът се извежда на един единствен ред, където се извежда името на цифрата на бълграски език. | |
| **Анализ на задачата**  С условната конструкция (**switch-case**) се сравняват стойностите на променливaта **number:**   * Ако променливата е цифра, се извежда еквалента на цифрата на български език. * Ако променливата не е цифра, се извежда съобщение за грешка. | |
| **Решение (сорс код)** | |
| using System;  {  class DigitToWord  {  static void Main(string[] args)  {  Console.WriteLine("Please input a one digit number:");  int number = int.Parse(Console.In.ReadLine());  switch (number)  {  case 1: Console.WriteLine("Едно"); break;  case 2: Console.WriteLine("Две"); break;  case 3: Console.WriteLine("Три"); break;  case 4: Console.WriteLine("Четири"); break;  case 5: Console.WriteLine("Пет"); break;  case 6: Console.WriteLine("Шест"); break;  case 7: Console.WriteLine("Седем"); break;  case 8: Console.WriteLine("Осем"); break;  case 9: Console.WriteLine("Девет"); break;  case 0: Console.WriteLine("Нула"); break;  default: Console.WriteLine("Please, input valid digits - from 0 to 9."); break;  }  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Примерният вход от условието на задачата. * Входните данни не са цифра – тогава се извежда съобщение за грешка. | |
| **Вход** | **Изход** |
| 5 | Пет |
| **Вход** | **Изход** |
| 10 | Please, input valid digits - from 0 to 9. |

|  |  |
| --- | --- |
| Задача 6. Решаване на квадратно уравнение | |
| **Условие**  Напишете програма, която при въвеждане на коефициентите (a, b и c) на квадратно уравнение: ax2+bx+c, изчислява и извежда неговите реални корени (ако има такива). Квадратните уравнения могат да имат 0, 1 или 2 реални корена. | |
| **Описание на входа**  Въвеждат се три числа, всяко на нов ред. Сътветно **a**, **b**, **c**. | |
| **Описание на изхода**  Ако квадратното уравнение няма решение, се изписва  No real roots  Ако има решение, се изписват неговите корени. | |
| **Анализ на задачата**  Изчислява се детерминаната и се записва в променливата **D**.  Ако тя е по-малка от нула, на конзолата се изписва „No real roots” и програмата прекратява решението си.  Ако детерминаната е равна на **0**, се изчислява единсвеният корен в променливата **root** и се изписва на конзолата.  Ако детерминаната е по-голяма от **0**, се изчисляват двата възможни корена и изписват на конзолата. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class SolvingEquation  {  static void Main(string[] args)  {  //on the 3 input lines are the coefficients: a b and c  double a = double.Parse(Console.ReadLine());  double b = double.Parse(Console.ReadLine());  double c = double.Parse(Console.ReadLine());  double D = b \* b - 4 \* a \* c; //finding the determinanta(Don't know what it is in English ;d)  //On the output line should be one of the three cases  //first case: "No real roots"  //second case: The only root  //this case: The 2 roots seperated by space  if (D < 0)  {  Console.WriteLine("No real roots");  }  if (D == 0)  {  double root = -b / 2 \* a;  Console.WriteLine(root);  }  else  {  Console.WriteLine((-b + Math.Sqrt(D)) / (2 \* a) + " " + (-b - Math.Sqrt(D)) / (2 \* a));  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * 4 2 0 -> 0 -0.5 * 0 0 0 -> 0 | |
| **Вход** | **Изход** |
| 4  2  0 | 0  -0.5 |
| **Вход** | **Изход** |
| 0  0  0 | 0 |
| **Вход** | **Изход** |
| -53  -98  -4 | -4  -53  -98 |

|  |  |
| --- | --- |
| Задача 7. Извеждане на най-голямото от пет числа | |
| **Условие**  Напишете програма, която намира най – голямото по стойност число, измежду пет дадени числа  Пример: 19 , 4 , **72** , 14, -25 🡪 **72** | |
| **Описание на входа**  Входните данни се въвеждат на пет реда, като на първия ред се въвежда първата променлива, на на втория ред се възвежда втората променлива, и аналогично на третия ред се възвежда третата променлива , четвъртия – четвъртата променлива и на петия – петата променлива. | |
| **Описание на изхода**  Изходът се извежда на един единствен ред, където се извежда най- голямото от петте числа. | |
| **Анализ на задачата**  Декларират се две помощни променливи **max1** и **max2**.   * **max1** приема стойността на по- голямата променлива от **firstNumber** и **secondNumber**, чрез използване на метода **Math.Max**(така намираме по- голямото от първите две числа). Метода **Math.Max** извежда по- голямата от две въведени стойностти. * **max2** приема стойността на по- голямата променлива от **forthNumber** и **fifthNumber**, чрез използване на метода **Math.Max**(така намираме по- голямото от последните две числа). * В следващата стъпка **max1**, отново чрез изпозване на **Math.Max** приема по-голямата стойност от променливата **thirdNumber** и старата си стойност.(След тази стъпка сме свели сравнението до две стойности) * В последната стъпка, отново се използва метода **Math.Max**, където **max1** приема по-голямата стойност от променливата **max2** и предишната си стойност.(В тази стъпка **max1** примема най- голямата стойност от петте зададени променливи) | |
| **Решение (сорс код)** | |
| using System;  {  class BiggestOfFiveNumbers  {  static void Main(string[] args)  {  Console.WriteLine("Please input a double for a:");  double firstNumber = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a double for b:");  double secondNumber = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a double for c:");  double thirdNumber = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a double for d:");  double forthNumber = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a double for e:");  double fifthNumber = double.Parse(Console.In.ReadLine());  double max1 = Math.Max(firstNumber, secondNumber);  double max2 = Math.Max(fifthNumber, forthNumber);  max1 = Math.Max(max1, thirdNumber);  max1 = Math.Max(max1, max2);  Console.WriteLine("The max number is:" + max1);  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Примерният вход от условието на задачата * Две равни, най - големи стойностти * Три равни, най - големи стойностти * Четири равни, най - големи стойностти * Пет равни, най - големи стойностти | |
| **Вход** | **Изход** |
| 19  4  **72**  14  -25 | 72 |
| **Вход** | **Изход** |
| 9  **505**  505  19  406 | 505 |
| **Вход** | **Изход** |
| -1088  **12334**  **12334**  **12334**  4 | 12334 |
| **Вход** | **Изход** |
| **-49203**  **-49203**  **-49203**  **-49203**  -500002 | -49203 |
| **Вход** | **Изход** |
| 42  42  42  42  42 | 42 |

|  |  |
| --- | --- |
| Задача 8. Прочитане на различен вид променлива | |
| **Условие**  Напишете програма, която по избор на потребителя прочита от конзолата променлива от тип int, double или string. Ако променливата е int или double, трябва да се увеличи с 1. Ако променливата е string, трябва да се прибави накрая символа "\*". Отпечатайте получения резултат на конзолата. Използвайте switch конструкция. | |
| **Описание на входа**  Потребителят въвежда на първия ред **int**, **double** или **string**. На вторият ред се въвежда съответното число или символен низ. | |
| **Описание на изхода**  Ако е въведено число, то се изписва на конзолата, събрано с числото 1. Ако е въведен символен низ, той се изписва на конзолата с добавена символа **\*** на края. | |
| **Анализ на задачата**  В променливата x се прочита първия ред от потребителя. Чрез switch конструкция се прави проверка дали той е въвел **int**, **double** или **string.** В първите два случая се очаква като вход на втория ред самото число, след което към него се добавя 1 и се отпечатва на конзолата.  В случай, че изборът на потребителя е **string**, символният низ се прочита и се изкарва на конзолата с прикрепен символ **\*** на края. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class AdjustingIntDoubleOrString  {  static void Main(string[] args)  {  //on the first input line should be a string {int,double,string} defining the type of the variable  // on the second input line should be the variable  // on the output line should be the result after the switch according to the rules  //if int or double add 1 if string concatenate with "\*"  string x = Console.ReadLine();  switch (x)  {  case "int":  int i = int.Parse(Console.ReadLine());  i = i + 1;  Console.WriteLine(i); break;  case "double":  double d = double.Parse(Console.ReadLine());  d = d + 1;  Console.WriteLine(d); break;  case "string":  string s = Console.ReadLine();  Console.WriteLine(s + "\*"); break;  default: ; break;  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Целочислено число – 442424 - 442425 * Дробно число – 554.34241 – 555.34241 * Текст – “Geogri Martin Georgi Ceco” -“Geogri Martin Georgi Ceco\*” * (празни данни) - (празни данни) | |
| **Вход** | **Изход** |
| 442424 | 442425 |
| **Вход** | **Изход** |
| 554.34241 | 555.34241 |
| **Вход** | **Изход** |
| Geogri Martin Georgi Ceco | Geogri Martin Georgi Ceco\* |
| **Вход** | **Изход** |
| (празни данни) | (празни данни) |

|  |  |
| --- | --- |
| Задача 9. Извеждане на подмножество равно на нула | |
| **Условие**  Дадени са пет цели числа. Напишете програма, която намира онези подмножества от тях, които имат сума **0**. Примери:   * Ако са дадени числата {3, **-2, 1, 1**, 8}, сумата на -2, 1 и 1 е 0. * Ако са дадени числата {3, 1, -7, 35, 22}, няма подмножества със сума 0. | |
| **Описание на входа**  Входът се състои от 5 цели числа. Всяко от тях е въведено на нов ред. | |
| **Описание на изхода**  Изходът се състои от един ред. Ако е намерено подмножество | |
| **Анализ на задачата**  Въвеждат се петте числа, като всяко се конвертира от тип string в тип double и се записва в масива **numbers**. Ако при въвеждането потребителят направи грешка, въвеждайки символ, различен от цифра или прекалено дълго число, се прихваща **FormatException** или **OverflowException**, програмата прекратява изпълнението си и се показва видът на грешката.  Проверката се състои от 2 вложени цикъла. Целта е да се проверяват сумите на числата, започвайки от първото до края, след това от второто до края и така докато не се изредят всички подмножества. Първият цикъл поставя променливата **startPos** на позиция **0** и всеки път я придвижва напред. Вторият започва от текущата позиция на **startPos** и премества променливата **endPos** до края на масива с числа.  При всяко придвижване на **startPos** една позиция напред, сумата се нулира, за да може вторият цикъл да сметне наново сума, която евентуално да стане **0**.  На всяка стъпка се изчислява сумата sum и ако тя стане равна на **0**, с един цикъл се показват всички числа между позиции **startPos** и **endPos** в масива. Това са числата, чиято сума е **0**.  Ако бъде открито поне едно подмножество, булевата променлива **subsetFound** приема стойност **true**. Ако на края **subsetFound** имат стойност **false**, се изписва на конзолата, че не е намерено нито едно подмножество, равно на **0**. | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class Task9  {  static void Main(string[] args)  {  const int NUMBERS\_COUNT = 5;  double[] number = new double[5];  bool subsetFound = false;  try  {  Console.WriteLine("Please input a number for a:");  number[0] = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a number for b:");  number[1] = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a number for c:");  number[2] = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a number for d:");  number[3] = double.Parse(Console.In.ReadLine());  Console.WriteLine("Please input a number for e:");  number[4] = double.Parse(Console.In.ReadLine());  }  catch (FormatException)  {  Console.WriteLine("Please, input valid number values.");  return;  }  catch (OverflowException)  {  Console.WriteLine("Please, input valid number.");  return;  }  for (int startPos = 0; startPos < NUMBERS\_COUNT; startPos++)  {  double sum = 0;  for (int endPos = startPos; endPos < NUMBERS\_COUNT; endPos++)  {  sum += number[endPos];  if (sum == 0)  {  Console.Write("Subset found: ");  subsetFound = true;  for (int iterator = startPos; iterator <= endPos; iterator++)  {  Console.Write("{0} ", number[iterator]);  }  Console.WriteLine();  }  }  }  if (subsetFound == false)  {  Console.WriteLine("No subset found.");  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Първият примерният вход от условието на задачата * Вторият примерният вход от условието на задачата * Всички входни данни са равни на нула – максимален резултат * Едната стойност е нула, и сбора на други две също е нула – три подмножества * Непоследователни стойностти равни на нула. | |
| **Вход** | **Изход** |
| 3  -2  1  1  8 | Subset found: -2 1 1 |
| **Вход** | **Изход** |
| 3  1  -7  35  22 | No subset found. |
| **Вход** | **Изход** |
| 0  0  0  0  0 | Subset found: 0  Subset found: 0 0  Subset found: 0 0 0  Subset found: 0 0 0 0  Subset found: 0 0 0 0 0  Subset found: 0  Subset found: 0 0  Subset found: 0 0 0  Subset found: 0 0 0 0  Subset found: 0  Subset found: 0 0  Subset found: 0 0 0  Subset found: 0  Subset found: 0 0  Subset found: 0 |
| **Вход** | **Изход** |
| -1  1  0  5  4 | Subset found: -1 1  Subset found: -1 1 0  Subset found: 0 |
| **Вход** | **Изход** |
| -5  57  63  14  5 | No subset found. |

|  |  |
| --- | --- |
| Задача 10. Прибавяне на бонус точки към число | |
| **Условие**  Напишете програма, която прилага бонус точки към дадени точки в интервала [1..9] чрез прилагане на следните правила:   * Ако точките са между 1 и 3, програмата ги умножава по 10. * Ако точките са между 4 и 6, ги умножава по 100. * Ако точките са между 7 и 9, ги умножава по 1000. * Ако точките са 0 или повече от 9, се отпечатва съобщение за грешка. | |
| **Описание на входа**  Потребителят въвежда на първия ред едно единствено число. | |
| **Описание на изхода** | |
| **Анализ на задачата**  Въведеното число се запазва в променливата **digit**. Следва try клауза. Ако кодът, изпъплнен в нея хвърли **exception**, то той ще бъде прихванат в catch клаузата по-долу.  Използва се оператор **switch**, който според съдържанието на променливата **x**, предприема следните действия:   * **0** или **9**: хвърля се exception, защото това е недопустим вход. * **1-3**: числото се умножава по **10** и резултатът се изписва на конзолата. * **4-6**: числото се умножава по **100** и резултатът се изписва на конзолата. * **7-9**: числото се умножава по **1000** и резултатът се изписва на конзолата.   Ако е бил хвърлен **exception**, той се прихваща в клаузата **catch** и се изписва съобщение за грешка. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  class AddingBonusPointsToAdigit  {  static void Main(string[] args)  {  //On the first input line is a digit (0-9)  //On the output line is one of the three cases  //if the digit is 0, 9 or more an exception is thrown and the output line is "Bad Input!!!"  //if the digit is 1, 2 or 3 the output is the digit multiplied by 10  //if the digit is 4 5 or 6 the output is the digit multiplied by 100  //if the digit is 7 8 or 9 the output is the digit multiplied by 1000  int digit = int.Parse(Console.ReadLine());  try  {  switch (digit)  {  case 0:  throw new Exception("Bad Input!!!");  break;  case 1:  case 2:  case 3:  digit = digit \* 10;  Console.WriteLine(digit);  break;  case 4:  case 5:  case 6:  digit = digit \* 100;  Console.WriteLine(digit);  break;  case 7:  case 8:  case 9:  digit = digit \* 1000;  Console.WriteLine(digit);  break;  default:  throw new Exception("Bad Input!!!");  break;  }  }  catch (Exception ex)  {  Console.WriteLine(ex.Message);  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * 1 – 10 * 5 – 500 * 8 – 8000 * 0 – Bad Input!!! * 14 – Bad Input!!! | |
| **Вход** | **Изход** |
| 1 | 10 |
| **Вход** | **Изход** |
| 5 | 500 |
| **Вход** | **Изход** |
| 8 | 8000 |
| **Вход** | **Изход** |
| 0 | Bad Input!!! |
| **Вход** | **Изход** |
| 14 | Bad Input!!! |
| **Вход** | **Изход** |
| (празни данни) | Bad Input!!! |

|  |  |
| --- | --- |
| Задача 11. Извеждане на името на число | |
| **Условие**  Напишете програма, която за преобразува дадено число в интервала [0…999] в текст, съответстващ на българското произношение на числото.  Примери:   * 0 🡪 Нула * 12 🡪 Дванайсет * 98 🡪 Деветдесет и осем * 273 🡪 Двеста седемсет и три * 400 🡪 Четиристотин * 501 🡪 Петстотин и едно * 711 🡪 Седемстотин и единайсет | |
| **Описание на входа**  Входните данни се въвеждат на един единствен ред. | |
| **Описание на изхода**  Изходът се извежда на един единствен ред, където се извежда името на числото на бълграски език. | |
| **Анализ на задачата**  Числото се въвежда в променливата **param**. Ако то е между 0 и 999 включително, се дели на 100 и резултатът от 0 до 9 се проверява в **switch**, чрез който се изважда на конзолата първата дума, съответстваща на стотици.  Ако числото не е кръгло, т.е. има остатък при делене на 100 и е различно от 0, на конзолата се извежда „ и „. Следва **switch**, който проверява резултата от разделянето на числото на 10 и остатъка от второ делене на 10.   * Ако той е **кръгъл** **(без остатък)**, се изписва последната дума, която е според числото е например „десет” или „единадесет”. * Ако **не е кръгъл** **(има остатък)**, се изписват десетиците и съюзът „ и” . Например „трийсет и”.   На края според остатъка от делене на 10 на променливата **temp**, се завършва последната част от числото или единиците. Например „едно”. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace  {  class NumberToWords  {  static void Main(string[] args)  {  int param = 0;    Console.WriteLine("Please enter a number:");  param = int.Parse(Console.In.ReadLine());    if (param <= 999 && param >= 0)  {  int temp = param;  switch (temp / 100)  {  case 0:  break;  case 1:  Console.Write("Сто ");  break;  case 2:  Console.Write("Двеста ");  break;  case 3:  Console.Write("Триста ");  break;  case 4:  Console.Write("Четиристотин ");  break;  case 5:  Console.Write("Петстотин ");  break;  case 6:  Console.Write("Шестстотин ");  break;  case 7:  Console.Write("Седемстотин ");  break;  case 8:  Console.Write("Осемстотин ");  break;  case 9:  Console.Write("Деветстотин ");  break;  default: Console.WriteLine("Error report!"); break;  }  if (temp / 100 != 0 && temp % 100 != 0)  {  Console.Write("и ");  }  switch (temp / 10 % 10)  {  case 0:  break;  case 1:  {  switch (temp % 10)  {  case 0:  Console.WriteLine("десет ");  break;  case 1:  Console.WriteLine("единадесет ");  break;  case 2:  Console.WriteLine("дванайсет ");  break;  case 3:  Console.WriteLine("тринайсет ");  break;  case 4:  Console.WriteLine("четиранайсет ");  break;  case 5:  Console.WriteLine("петнайсет ");  break;  case 6:  Console.WriteLine("шестнайсет ");  break;  case 7:  Console.WriteLine("седемнайсет ");  break;  case 8:  Console.WriteLine("осемнайсет ");  break;  case 9:  Console.WriteLine("деветнайсет ");  break;  default: Console.WriteLine("Error report!"); break;  }  }  break;  case 2:  Console.Write("двайсет и ");  break;  case 3:  Console.Write("трийсет и ");  break;  case 4:  Console.Write("четирсет и ");  break;  case 5:  Console.Write("педесет и ");  break;  case 6:  Console.Write("шейсет и ");  break;  case 7:  Console.Write("седемдесет и ");  break;  case 8:  Console.Write("осемдесет и ");  break;  case 9:  Console.Write("деветдесет и ");  break;  default: Console.Write("Error report!"); break;  }  switch (temp % 10)  {  case 0:  if (temp == 0)  {  Console.Write("нула");  }  Console.Write("\n ");  break;  case 1:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("едно");  break;  case 2:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("две");  break;  case 3:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("три ");  break;  case 4:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("четири");  break;  case 5:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("пет");  break;  case 6:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("шест");  break;  case 7:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("седем ");  break;  case 8:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("осем ");  break;  case 9:  if (temp / 10 % 10 == 1) break;  Console.WriteLine("девет ");  break;  default: Console.WriteLine("Error report!"); break;  }  }  else  {  Console.WriteLine(" Out of range! ");  }  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Примерният вход от условието на задачата. * Число извън зададени интервал   🡪   * 🡪 🡪 * 🡪 * 🡪 * 🡪 | |
| **Вход** | **Изход** |
| 0 | Нула |
| **Вход** | **Изход** |
| 12 | Дванайсет |
| **Вход** | **Изход** |
| 98 | Деветдесет и осем |
| **Вход** | **Изход** |
| 273 | Двеста седемсет и три |
| **Вход** | **Изход** |
| 400 | Четиристотин |
| **Вход** | **Изход** |
| 501 | Петстотин и едно |
| **Вход** | **Изход** |
| 711 | Седемстотин и единайсет |
| **Вход** | **Изход** |
| -5 | Out of range! |