# 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

**Александър Михайлов Дженков - 70\_ALDJ –** [**a.djenkov90@gmail.com**](mailto:a.djenkov90@gmail.com)**, phone: 0883486223**

**Кристиан Димитров Димитров -223\_KRDI –** [**kristian.dimitrov.sz@gmail.com**](mailto:kristian.dimitrov.sz@gmail.com)**,p: 0883540544**

**Александър Владиславов Лазаров-253\_ALLA –** [**sass\_waz@abv.bg**](mailto:sass_waz@abv.bg)**, phone: 0883478803**

**Здравко Йорданов Брънзов - 14\_ZDBR –** [**z\_branzov@abv.bg**](mailto:z_branzov@abv.bg) **, phone: 0895700739**

|  |  |
| --- | --- |
| **Задача 1. Swap Integers** | |
| **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. | |
| **Описание на входа**  Входните данни са две числа въведени едно след друго на два отделни реда. | |
| **Описание на изхода**  Изходът се извежда на два реда с по едно число всеки. При правилно написана програма по-малкото от двете въведени числа би трябвало да е първо. | |
| **Анализ на задачата**  Проверяваме с един оператор (**if**) дали първото число е по-голямо от второто. Ако е използваме временна променлива, на която присвояваме първата стойност **temp = var1**. След това на присвояваме на променливата **var1 = var2, а на var2 = temp**. Изписваме изхода в конзолата. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace SwapTwoIntegersTask1  {  class SwapTwoIntegersTask1  {  static void Main(string[] args)  {  int var1 = int.Parse(Console.ReadLine());  int var2 = int.Parse(Console.ReadLine());  int temp=0;  if (var1 > var2)  {  temp = var1;  var1 = var2;  var2 = temp;  }  Console.WriteLine("{0}\n{1}", var1, var2);  }  }  } | |
| **Тестове**  Решението е тествано с номинални случаи, крайни числа, отрицателни и положителни числа, нула, плюс на числа и нули пред числата. Равенство между числа, обръщане на числа, когато едното е по-голямо от другото. | |
| **Вход** | **Изход** |
| 2  3 | 2  3 |
| **Вход** | **Изход** |
| 6  4 | 4  6 |
| **Вход** | **Изход** |
| 100  100 | 100  100 |
| **Вход** | **Изход** |
| 2147483647  -2147483648 | -2147483648  2147483647 |
| **Вход** | **Изход** |
| -123456  -123456 | 123456  123456 |
| **Вход** | **Изход** |
| -0  0 | 0  0 |
| **Вход** | **Изход** |
| 00002  00001 | 00001  00002 |
| **Вход** | **Изход** |
| 000032  00000024 | 00000024  000032 |
| **Вход** | **Изход** |
| +12345  -12345 | -12345  12345 |
| **Вход** | **Изход** |
| +000000  -000001 | -1  0 |

|  |  |
| --- | --- |
| **Задача 2. Sign of a product of three numbers** | |
| **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 три числа последователно на отделен ред всяко. Входа не съдържа числото нула. | |
| **Описание на изхода**  Изходът представлява един ред на който се печата знака на произведението. | |
| **Анализ на задачата**  При въвеждането на три числа от страна на потребителя се създават три променливи **(x,y,z)**, на които се присвояват стойностите. С конструкцията **if else** сравняваме трите променливи с нула и намираме дали са положителни или отрицателни :   * **x > 0 && y > 0 && z > 0** * **x < 0 && y < 0 && z > 0** * **x > 0 && y < 0 && z < 0** * **x < 0 && y > 0 && z < 0**   Спрямо резултатите получаваме и отговора, който се принтира от тялото на **if-a**. Ако и трите са положителни „+”, ако има две променливи от трите отрицателни „+” , при всички останали случаи „-” .Ако имаме нула в една от променливите изписва "It's zero result" . | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_02SignOfProductOfThreeRealNum  {  class Program  {  static void Main(string[] args)  {  double x = double.Parse(Console.ReadLine());  double y = double.Parse(Console.ReadLine());  double z = double.Parse(Console.ReadLine());  if (x > 0 && y > 0 && z > 0)  {  Console.WriteLine("+");  }  else if (x < 0 && y < 0 && z > 0)  {  Console.WriteLine("+");  }  else if (x > 0 && y < 0 && z < 0)  {  Console.WriteLine("+");  }  else if (x < 0 && y > 0 && z < 0)  {  Console.WriteLine("+");  }  else if (x == 0 || y == 0 || z == 0)  {  Console.WriteLine("It's zero result");  }  else  {  Console.WriteLine("-");  }  }  }  } | |
| **Тестове**  Решението е тествано с номинални случаи, проверка на трите места с най-големи числа, проверка на отрицателни числа, на десетични, на нула, на повтарящи се числа. | |
| **Вход** | **Изход** |
| 0  1  -1 | It's zero result |
| **Вход** | **Изход** |
| 0  0  0 | It's zero result |
| **Вход** | **Изход** |
| -1000  +1000  -1000 | **+** |
| **Вход** | **Изход** |
| -0.01  -0.02  -0.03 | **-** |
| **Вход** | **Изход** |
| -0.0000001  -0.0000010  -0.0000100 | **-** |
| **Вход** | **Изход** |
| -0.1  1.11111  -99.999999 | **+** |
| **Вход** | **Изход** |
| 1000000000000000  -10000000000000  1000000000000000 | **-** |
| **Вход** | **Изход** |
| -999999999999999  -999999999999999  9999999999999999 | **+** |
| **Вход** | **Изход** |
| 0  9999999999999999999  -999999999999999999 | It's zero result |
| **Вход** | **Изход** |
| -100000000000000000000  0  0 | It's zero result |

|  |  |
| --- | --- |
| **Задача 3. Biggest of three numbers.** | |
| **Условие**  Напишете програма, която намира най-голямото по стойност число, измежду три дадени числа. | |
| **Описание на входа**  Въвеждат се три числа последователно на отделен ред всяко. | |
| **Описание на изхода**  Изходът представлява един ред на който се печата най-голямото число. | |
| **Анализ на задачата**  Присвоява се стойността на първото въведено число върху променливата **max**. След това се проверява с оператора (**if**) дали следващото число не е по – голямо от **max** ако да то на **max**  се присвоява тази стойност. Операцията се повтаря още веднъж и за следващото число.  Отпечатваме максималното число **max**. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace BiggestFromThreeTask3  {  class BiggestFromThreeTask3  {  static void Main(string[] args)  {  decimal var1 = decimal.Parse(Console.ReadLine());  decimal var2 = decimal.Parse(Console.ReadLine());  decimal var3 = decimal.Parse(Console.ReadLine());  decimal max = var1;  if (var2 > max)  {  max = var2;  }  if (var3 > max)  {  max = var3;  }  Console.WriteLine(max);  }  }  } | |
| **Тестове**  Решението е тествано с номинални случаи, проверка на трите места с най-големи числа, проверка на отрицателни числа, на десетични, на нула, на повтарящи се числа. | |
| **Вход** | **Изход** |
| 2  1  3 | 3 |
| **Вход** | **Изход** |
| 1234  12345  1232 | 12345 |
| **Вход** | **Изход** |
| 100000  10000  100 | 100000 |
| **Вход** | **Изход** |
| 1234567890  1234567890  1234567890 | 1234567890 |
| **Вход** | **Изход** |
| -0.01  -0.001  -0.0001 | -0.0001 |
| **Вход** | **Изход** |
| 5003343  +5003343  5003342 | 5003343 |
| **Вход** | **Изход** |
| .0000000000000000000000000000000033252  0.0000000000000000000000000000000000342  -0.000000000000000000000000000000000342 | 0.0000000000000000000000000000000033252 |
| **Вход** | **Изход** |
| 99.99999999999999999999  88.88888888888888888888  77.77777777777777777777 | 99.99999999999999999999 |
| **Вход** | **Изход** |
| -200000000000  -20.222222220  -111111111110 | -20.222222220 |
| **Вход** | **Изход** |
| 0.015  0.01500  0.0150000 | 0.015 |

|  |  |
| --- | --- |
| **Задача 4. Biggest of three numbers.** | |
| **Условие**  Сортирайте 3 реални числа в намаляващ ред. Използвайте вложени if оператори. | |
| **Описание на входа**  Въвеждат се три числа последователно на отделен ред всяко. | |
| **Описание на изхода**  Изходът представлява три реда, на които са изписани числата в намаляващ ред. | |
| **Анализ на задачата**  За решението се използват три **if** оператора. Всяко число се сравнява с всяко и ако второто е по-голямо си сменят местата и така три пъти.  Накрая се изписва изходът. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace SortIntsDescendingTask4  {  class SortIntsDescendingTask4  {  static void Main(string[] args)  {  double var1 = double.Parse(Console.ReadLine());  double var2 = double.Parse(Console.ReadLine());  double var3 = double.Parse(Console.ReadLine());  double min = var1;  double temp=0;  if (var2 >var1)  {  temp = var1;  var1 = var2;  var2 = temp;  }  if (var3 > var1)  {  temp = var1;  var1 = var3;  var3 = temp;  }  if (var3 > var2)  {  temp = var3;  var3 = var2;  var2 = temp;  }  Console.WriteLine("{0},{1},{2}", var1, var2, var3);  }  }  } | |
| **Тестове**  Решението е тествано с номинални случаи, проверка на трите места с най-големи числа, проверка на отрицателни числа, на десетични, на нула, на повтарящи се числа. | |
| **Вход** | **Изход** |
| 2  1  3 | 3  2  1 |
| **Вход** | **Изход** |
| 1  2  3 | 3  2  1 |
| **Вход** | **Изход** |
| 100000  10000  100 | 100000  10000  100 |
| **Вход** | **Изход** |
| 1234567890  1234567890  1234567890 | 1234567890  1234567890  1234567890 |
| **Вход** | **Изход** |
| 0.3333333  0.333333343267441  0.333333352143252 | 0.333333352143252  0.333333343267441  0.3333333 |
| **Вход** | **Изход** |
| 50033436565  +50033436565  50033426565 | 50033436565  50033436565  50033426565 |
| **Вход** | **Изход** |
| .0000000000000000000000000000000033252  0.0000000000000000000000000000000000342  -0.000000000000000000000000000000000342 | 3.3252E-33  3.42E-35  -3.42E-34 |
| **Вход** | **Изход** |
| 123456.00007899  123456.00007900  123456.00007898 | 123456.000079  123456.00007899  123456.00007898 |
| **Вход** | **Изход** |
| -200000000000  -20.222222220  -111111111110 | -20.222222220  -111111111110  -200000000000 |
| **Вход** | **Изход** |
| 0.099  0.09900  0.0990000 | 0.099  0.099  0.099 |

|  |  |
| --- | --- |
| **Задача 5. Translate digit in Bulgarian.** | |
| **Условие**  Напишете програма, която за дадена цифра (0-9), зададена като вход, извежда името на цифрата на български език. | |
| **Описание на входа**  Въвежда се само една цифра. | |
| **Описание на изхода**  Изписва превода на числото на български език на самостоятелен ред с малки букви. | |
| **Анализ на задачата**  Използва се един switch с десет опции за всяка цифра, който отпечатва резултата. Ако е въведен невалиден вход изписва „грешка”. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace TranslateZeroToNineTask5  {  class TranslateZetoToNineTask9  {  static void Main(string[] args)  {  byte digit = byte.Parse(Console.ReadLine());  switch (digit)  {  case 0: Console.WriteLine("нула"); break;  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;  default: Console.WriteLine("грешка"); break;  }  }  }  } | |
| **Тестове**  Решението е тествано с числата от нула до девет. | |
| **Вход** | **Изход** |
| 0 | нула |
| **Вход** | **Изход** |
| 1 | едно |
| **Вход** | **Изход** |
| 2 | две |
| **Вход** | **Изход** |
| 3 | три |
| **Вход** | **Изход** |
| 4 | четири |
| **Вход** | **Изход** |
| 5 | пет |
| **Вход** | **Изход** |
| 6 | шест |
| **Вход** | **Изход** |
| 7 | седем |
| **Вход** | **Изход** |
| 8 | осем |
| **Вход** | **Изход** |
| 9 | девет |

|  |  |
| --- | --- |
| **Задача 6. Quadratic Equation** | |
| **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**  Входът се състои от три числа на отделен ред. | |
| **Описание на изхода**  Изходът представлява един или повече редове в зависимост от броя корени. При изписана „0” значи няма корени. | |
| **Анализ на задачата**  Задачата е представена чрез условна конструкция с три **if**-a за всяка една от трите възможности на квадратното уравнение:   * да има един корен * да има два корена * да няма корени   При всяка възможност се изчислява дискриминантата и спрямо нея се пресмятат корените и съответно се изписват. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace QuadraticEquationTask6  {  class QuadraticEquationTask6  {  static void Main(string[] args)  {  Console.Write("Enter a:");  int a = int.Parse(Console.ReadLine());  Console.Write("Enter b:");  int b = int.Parse(Console.ReadLine());  Console.Write("Enter c:");  int c = int.Parse(Console.ReadLine());  double x1;  double x2;  int dis = (b \* b) - (4 \* a \* c);  if (dis > 0)  {  x1 = (-(double)b - Math.Sqrt(dis)) / (double)2 \* a;  x2 = (-(double)b + Math.Sqrt(dis)) / (double)2 \* a;  Console.WriteLine("{0}\n{1}", x1, x2);  }  if (dis == 0)  {  x1 = -(double)b / (double)2 \* a;  Console.WriteLine("{0}", x1);  }  if (dis < 0) Console.WriteLine("0");  }  }  } | |
| **Тестове**  Решението е тествано с отрицателни, положителни числа, нула, големи числа, обхванати са и трите възможни резултата от уравнението. | |
| **Вход** | **Изход** |
| 3  9  6 | -18  -9 |
| **Вход** | **Изход** |
| 1  -3  -4 | -1  4 |
| **Вход** | **Изход** |
| 1  1  1 | 0 |
| **Вход** | **Изход** |
| -38953587  -38953587  -38953587 | -757869926673917  -759434107318652 |
| **Вход** | **Изход** |
| 1  0  -9 | -3  3 |
| **Вход** | **Изход** |
| 1  0  500 | 0 |
| **Вход** | **Изход** |
| -400000000  1200000000  -900000000 | 2400000000 |
| **Вход** | **Изход** |
| 1234567890  1234567890  -1234567890 | -7.62103006994308E+17  -7.62054868024744E+17 |
| **Вход** | **Изход** |
| 898989  876543  123456 | 0 |
| **Вход** | **Изход** |
| -241459  -356634  -3253522 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Задача 7. Biggest of five numbers** | | | |
| **Условие**  Напишете програма, която намира най-голямото по стойност число измежду дадени 5 числа. | | | |
| **Описание на входа**  Входа е от пет различни числа всяко на отделен ред. | | | |
| **Описание на изхода**  Един ред на който се изписва най-голямото число. | | | |
| **Анализ на задачата**  Използват се пет променливи и конструкция **if else** при която се проверява пет пъти кое число е по-голямо от другите четири като се сравнява поотделно с останалите. В следствие го изписва. | | | |
| **Решение (сорс код)** | | | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace BiggestFromFiveTask7  {  class BiggestFromFiveTask7  {  static void Main(string[] args)  {  double var1 = double.Parse(Console.ReadLine());  double var2 = double.Parse(Console.ReadLine());  double var3 = double.Parse(Console.ReadLine());  double var4 = double.Parse(Console.ReadLine());  double var5 = double.Parse(Console.ReadLine());  if (var1 > var2 && var1 > var3 && var1 > var4 && var1 > var5) Console.WriteLine("{0}", var1);  else if (var2 > var1 && var2 > var3 && var2 > var4 && var2 > var5) Console.WriteLine("{0}", var2);  else if (var3 > var2 && var3 > var1 && var3 > var4 && var3 > var5) Console.WriteLine("{0}", var3);  else if (var4 > var2 && var4 > var3 && var4 > var1 && var4 > var5) Console.WriteLine("{0}", var4);  else if (var5 > var2 && var5 > var3 && var5 > var4 && var5 > var1) Console.WriteLine("{0}", var5);    }  }  } | | | |
| **Тестове**  Решението е тествано с номинални случаи, проверка на петте места с най-големи числа, проверка на отрицателни числа, на десетични, на нула, на повтарящи се числа. | | | |
| **Вход** | | **Изход** |
| 0  3  2  7  4 | | 7 |
| **Вход** | | **Изход** |
| 100000  2234235  334235  456346  557457 | | 2234235 |
| **Вход** | | **Изход** |
| 9000000  800000  70000  60000  5000 | | 9000000 |
| **Вход** | | **Изход** |
| 34863  1325246  32523534  11111  111 | | 32523534 |
| **Вход** | | **Изход** |
| 11111111  88888888  77777777  44444444  99999999 | | 99999999 |
| **Вход** | | **Изход** |
| 123456  12345678900  123456789000  0987654321  09876543210 | | 123456789000 |
| **Вход** | | **Изход** |
| 01  10  011  110  0101 | | 110 |
| **Вход** | | **Изход** |
| 0.00000001  0.21352362  0.1353535234  0.1123231233  0.99 | | 0.99 |
| **Вход** | | **Изход** |
| -0.86848367  -0.487534872  -0.32543232  -0.014335533  -.01324 | -0.01234 |
| **Вход** | **Изход** |
| -10000  10000  1000  +10001  10002 | 10002 |

|  |  |
| --- | --- |
| **Задача 8. Choice of input data** | |
| **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**  Първият ред от входа трябва да е число от 0 до 2, което указва типа входни данни. Вторият ред е стойността на променливата в зависимост от избрания тип. | |
| **Описание на изхода**  Изходът е от един ред представляващ отговора вече обработен. При невалиден избор на данни изписва "No such а choice". | |
| **Анализ на задачата**  За решението е използван **switch** , чрез който се указва избора на данни като преди това опциите за избор са принтирани на екрана и са отбелязани с цифри от **0 до 2** .При грешно въведен избор извежда съобщение "No such а choice" . | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace IntDoubleStringTask8  {  class IntDoubleStringTask8  {  static void Main(string[] args)  {  Console.Write("0.Int\n1.Double\n2.string\nChoice:");  int choice = int.Parse(Console.ReadLine());  double number;  string myStr;  switch (choice)  {  case 0:  number = int.Parse(Console.ReadLine());  number++;  Console.WriteLine("{0}", number);  break;  case 1:  number = double.Parse(Console.ReadLine());  number++;  Console.WriteLine("{0}", number);  break;  case 2:  myStr = Console.ReadLine();  myStr = myStr + "\*";  Console.WriteLine("{0}", myStr);  break;  default:  Console.WriteLine("No such а choice");  break;  }  }  }  } | |
| **Тестове**  Решението на задачата обхваща проверка на кейс-а, дали добавя 1 на числата и дали слага \* след стринга. | |
| **Вход** | **Изход** |
| 0  2141234 | 2141235 |
| **Вход** | **Изход** |
| 0  -346534256 | -346534255 |
| **Вход** | **Изход** |
| 0  0000000001 | 2 |
| **Вход** | **Изход** |
| 1  0.0111111111 | 1.0111111111 |
| **Вход** | **Изход** |
| 1  -99999.9999 | -99998.9999 |
| **Вход** | **Изход** |
| 0  0.001 | 1.001 |
| **Вход** | **Изход** |
| 1  +1000234 | 1000235 |
| **Вход** | **Изход** |
| 2  Fjbkfsjdeg | fjbkfsjdeg\* |
| **Вход** | **Изход** |
| 2  After the text you must see a star | After the text you must see a star\* |
| **Вход** | **Изход** |
| 2  Jgrkjghrjkg hrkjgherjgkfhgkjdsfh sjkldfhdsjkbgshdkjfhdsjbgsdkjnsdjdngjdfsgbndfsjgbdsjfbds | Jgrkjghrjkg hrkjgherjgkfhgkjdsfh sjkldfhdsjkbgshdkjfhdsjbgsdkjnsdjdngjdfsgbndfsjgbdsjfbds\* |

|  |  |
| --- | --- |
| **Задача 9. Подмножества равни 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**  Въвеждат се пет цели числа на отделен ред всяко. | |
| **Описание на изхода**  Ако имаме подмножества програмата принтира за всяко подмножество неговите числа разделени с интервал **„ ”** и накрая завършващо с **„is 0”**. Примерно: „**2 -2 is 0**”. При множество с повтарящи се елементи трябва да излизат и повторенията. Пример: **{1, -1, 1, 5, 9}** => **1 -1 is 0 , -1 1 is 0** | |
| **Анализ на задачата**  В даденото решение задачата използва масив за съхранение на числата, но може да се създадът пет променливи и на тях да се присвоят стойностите. След това за всеки вариант сума от няколко числа се пише проверка с **if** дали не е равна на нула. Ако е, то тя се принтира и флага (int flagFound) за намерено подмножество се променя на равен на 1. Накрая ако няма нито едно намерено подмножество равно на нула то флага ще е равен на ‘0’ и следователно изписваме "No subset sum of 0 found" . | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace SubsetSumTask9  {  class SubsetSumTask9  {  static void Main(string[] args)  {  int[] array = new int[5];  array[0] = int.Parse(Console.ReadLine());  array[1] = int.Parse(Console.ReadLine());  array[2] = int.Parse(Console.ReadLine());  array[3] = int.Parse(Console.ReadLine());  array[4] = int.Parse(Console.ReadLine());  int flagFound = 0;  if ((array[0] + array[1]) == 0) { Console.WriteLine("{0} {1} is 0", array[0], array[1]); flagFound = 1; }  if ((array[0] + array[2]) == 0) { Console.WriteLine("{0} {1} is 0", array[0], array[2]); flagFound = 1; }  if ((array[0] + array[3]) == 0) { Console.WriteLine("{0} {1} is 0", array[0], array[3]); flagFound = 1; }  if ((array[0] + array[4]) == 0) { Console.WriteLine("{0} {1} is 0", array[0], array[4]); flagFound = 1; }  if ((array[1] + array[2]) == 0) { Console.WriteLine("{0} {1} is 0", array[1], array[2]); flagFound = 1; }  if ((array[1] + array[3]) == 0) {Console.WriteLine("{0} {1} is 0", array[1], array[3]);flagFound = 1; }  if ((array[1] + array[4]) == 0) {Console.WriteLine("{0} {1} is 0", array[1], array[4]);flagFound = 1; }  if ((array[2] + array[3]) == 0) {Console.WriteLine("{0} {1} is 0", array[2], array[3]);flagFound = 1; }  if ((array[2] + array[4]) == 0) {Console.WriteLine("{0} {1} is 0", array[2], array[4]);flagFound = 1; }  if ((array[3] + array[4]) == 0) {Console.WriteLine("{0} {1} is 0", array[3], array[4]);flagFound = 1; }  if ((array[0] + array[1] + array[2]) == 0){ Console.WriteLine("{0} {1} {2} is 0", array[0], array[1], array[2]);flagFound = 1; }  if ((array[0] + array[1] + array[3]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[0], array[1], array[3]);flagFound = 1; }  if ((array[0] + array[1] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[0], array[1], array[4]);flagFound = 1; }  if ((array[0] + array[2] + array[3]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[0], array[2], array[3]);flagFound = 1; }  if ((array[0] + array[2] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[0], array[2], array[4]);flagFound = 1; }  if ((array[0] + array[3] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[0], array[3], array[4]);flagFound = 1; }  if ((array[1] + array[2] + array[3]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[1], array[2], array[3]);flagFound = 1; }  if ((array[1] + array[2] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[1], array[2], array[4]);flagFound = 1; }  if ((array[1] + array[3] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[1], array[3], array[4]);flagFound = 1; }  if ((array[2] + array[3] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} is 0", array[2], array[3], array[4]);flagFound = 1; }  if ((array[0] + array[1] + array[2] + array[3]) == 0){ Console.WriteLine("{0} {1} {2} {3} is 0", array[0], array[1], array[2], array[3]);flagFound = 1; }  if ((array[0] + array[1] + array[2] + array[4]) == 0){ Console.WriteLine("{0} {1} {2} {3} is 0", array[0], array[1], array[2], array[4]);flagFound = 1; }  if ((array[0] + array[2] + array[3] + array[4]) == 0){ Console.WriteLine("{0} {1} {2} {3} is 0", array[0], array[2], array[3], array[4]);flagFound = 1; }  if ((array[1] + array[2] + array[3] + array[4]) == 0){ Console.WriteLine("{0} {1} {2} {3} is 0", array[1], array[2], array[3], array[4]);flagFound = 1; }  if ((array[0] + array[1] + array[2] + array[3] + array[4]) == 0) {Console.WriteLine("{0} {1} {2} {3} {4} is 0", array[0], array[1], array[2], array[3], array[4]);flagFound = 1; }  if (flagFound == 0) Console.WriteLine("No subset sum of 0 found");  }  }  } | |
| **Тестове**  Тестването на програмата обхваща въвеждане на пет поредни числа, след което проверява сбора на всяко с всяко и всички възможни 26 комбинации като се изключват вече проверените. | |
| **Вход** | **Изход** |
| 0  0  0  0  0 | 0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 is 0  0 0 0 0 is 0  0 0 0 0 is 0  0 0 0 0 is 0  0 0 0 0 is 0  0 0 0 0 0 is 0 |
| **Вход** | **Изход** |
| 1  1  1  1  1 | No subset sum of 0 found |
| **Вход** | **Изход** |
| 1  -1  2  -2  69 | 1 -1 is 0  2 -2 is 0  1 -1 2 -2 is 0 |
| **Вход** | **Изход** |
| -99  90  9  0  2 | -99 90 9 is 0  -99 90 9 0 is 0 |
| **Вход** | **Изход** |
| -1  -1  -1  -1  4 | -1 -1 -1 -1 4 is 0 |
| **Вход** | **Изход** |
| 2  -2  4  0  -4 | 2 -2 is 0  4 -4 is 0  2 -2 0 is 0  4 0 -4 is 0  2 -2 4 -4 is 0  2 -2 4 0 -4 is 0 |
| **Вход** | **Изход** |
| -1000  1  1001  -1002  1000 | -1000 1000 is 0  1 1001 -1002 is 0  -1000 1 1001 -1002 1000 is 0 |
| **Вход** | **Изход** |
| 54  -98  88  -65  -17 | No subset sum of 0 found |
| **Вход** | **Изход** |
| -1  -2  -3  -4  -5 | No subset sum of 0 found |
| **Вход** | **Изход** |
| 2  3  -2  -3  2 | 2 -2 is 0  3 -3 is 0  -2 2 is 0  2 3 -2 -3 is 0 |

|  |  |
| --- | --- |
| **Задача 10. Bonus Points** | |
| **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**  Входа е един ред с една цифра. | |
| **Описание на изхода**  Ред с резултата ако входа е верен или съобщение за грешка "Error" . | |
| **Анализ на задачата**  Използва се проверка в началото дали входа е число. След това със **switch** се проверява дали стойността е измежду зададените в условието. Ако да, обработва я и я изписва. Ако не, изписва съобщение за грешка "Error". | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace AddBonusPontsTask10  {  class AddBonusPontsTask10  {  static void Main(string[] args)  {  string x = Console.ReadLine();  int y = 0;  if (!int.TryParse(x, out y))  {  Console.WriteLine("Error");  return;  }  switch (y)  {  case 1:  case 2:  case 3:  Console.WriteLine(y \* 10);  break;  case 4:  case 5:  case 6:  Console.WriteLine(y \* 100);  break;  case 7:  case 8:  case 9:  Console.WriteLine(y \* 1000);  break;  default: Console.WriteLine("Error");  break;  }  }  }  } | |
| **Тестове**  Решението е тествано с числата от нула до девет. | |
| **Вход** | **Изход** |
| 0 | Error |
| **Вход** | **Изход** |
| 1 | 10 |
| **Вход** | **Изход** |
| 2 | 20 |
| **Вход** | **Изход** |
| 3 | 30 |
| **Вход** | **Изход** |
| 4 | 400 |
| **Вход** | **Изход** |
| 5 | 500 |
| **Вход** | **Изход** |
| 6 | 600 |
| **Вход** | **Изход** |
| 7 | 7000 |
| **Вход** | **Изход** |
| 8 | 8000 |
| **Вход** | **Изход** |
| 9 | 9000 |
| **Вход** | **Изход** |
| 10 | Error |

|  |  |
| --- | --- |
| **Задача 11.Translate number in Bulgarian** | |
| **Условие**  Напишете програма, която преобразува дадено число в интервала [0..999] в текст, съответстващ на българското произношение на числото. Примери:   * 0 → "Нула" * 12 → "Дванадесет" * 98 → "Деветдесет и осем" * 273 → "Двеста седемдесет и три" * 400 → "Четиристотин" * 501 → "Петстотин и едно" * 711 → "Седемстотин и единадесет" | |
| **Описание на входа**  Входа е един ред, на който се подава числото за превод. | |
| **Описание на изхода**  Изхода представлява превода на числото на български под формата на текст като всяко число започва с главна буква както и всяка негова част като единици, десетици и стотици. | |
| **Анализ на задачата**  Първо настройваме енкондинга на конзолата, за да може да пише на Български като и присвояваме енкодинг UTF8 (Console.OutputEncoding = Encoding.UTF8;). След това присвояваме числото на променливата, която сме създали и го раздробяваме на отделни променливи единици, десетици , стотни като го делим с остатък съответно първо на 10 да получим единиците след това делим на 10 , за да ги премахнем и пак делим с остатък за десетиците и накрая делим цялото число на 100 да получим стотиците. След това с помощта на switch за стотиците сравняваме стойноста и я изписваме при съвпадение. Правим проверка дали десетиците и единиците са равни на нула ако:   * да, пишем нов ред за по-приличен изход на отговора.   Продължаваме с проверка дали единиците са равни на нула + дали десетиците са различни от 1 и от 0:   * + ако да, добавяме **„и ”** към отговора   Преминаваме към **switch** за десетиците.   * Ако са равни на едно добавяме „и ” към отговора и продължаваме с вложен   + **switch** в който сравняваме единиците и при съвпадение изписваме числата от 10 до 19. * при цифри от 2-9 изписваме еквивалентните отговори за десетици.   И продължаваме с проверка if за десетиците. Ако са различни от 1 или 0 или ако стотиците и единиците са различни от нула добавяме „и ” към отговора. След това проверяваме с switch превода за единиците и го изписваме.  Накрая имаме проверка за случая ако е въведено числото „0” т.е. дали единиците, десетиците и стотиците са == на 0 при което изписваме „Нула”.  Ако няма въведено число в диапазона изписва "Out of range" . | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_11NumberPronunciation  {  class Program  {  static void Main(string[] args)  {  Console.OutputEncoding = Encoding.UTF8;  int x = int.Parse(Console.ReadLine());  int ones = x % 10;  int tens = (x / 10) % 10;  int hundrets = x / 100;  if(x>=0 && x<=999) {  switch (hundrets)  {  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;  case 0:  break;  }  if(ones == 0 && tens ==0)  Console.WriteLine();  if (ones == 0 && tens != 1 && tens!=0)  Console.Write("и ");  switch (tens)  {  case 1:  if (hundrets != 0 && (tens != 0 || ones > 0))  Console.Write("и ");  switch (ones)  {  case 1:  Console.WriteLine("Единадесет ");  break;  case 2:  Console.WriteLine("Дванадесет ");  break;  case 3:  Console.WriteLine("Тринадесет ");  break;  case 4:  Console.WriteLine("Четиринадесет ");  break;  case 5:  Console.WriteLine("Петнадесет ");  break;  case 6:  Console.WriteLine("Шеснадесет ");  break;  case 7:  Console.WriteLine("Седемнадесет ");  break;  case 8:  Console.WriteLine("Осемнадесет ");  break;  case 9:  Console.WriteLine("Деветнадесет ");  break;  case 0:  Console.WriteLine("Десет ");  break;  }  break;  case 2:  Console.Write("Двадесет ");  break;  case 3:  Console.Write("Тридесет ");  break;  case 4:  Console.Write("Четирдесет ");  break;  case 5:  Console.Write("Педесет ");  break;  case 6:  Console.Write("Шестдесет ");  break;  case 7:  Console.Write("Седемдесет ");  break;  case 8:  Console.Write("Осемдесет ");  break;  case 9:  Console.Write("Деветдесет ");  break;  case 0:  break;  }  if (tens != 1)  {  if(tens!=0 || (hundrets!=0 && ones!=0))  if (ones > 0)  {  Console.Write("и ");  }  switch (ones)  {  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:  if (tens != 0)  Console.WriteLine();  break;  }  }  if (ones == 0 && tens == 0 && hundrets == 0)  Console.WriteLine("Нула");  }  }else Console.WriteLine("Out of range");  }  } | |
| **Тестове**  Въвежда се прозволно число от 0-999 включително | |
| **Вход** | **Изход** |
| 0 | Нула |
| **Вход** | **Изход** |
| 999 | Деветстотин Деветдесет и Девет |
| **Вход** | **Изход** |
| 998 | Деветстотин Деветдесет и Осем |
| **Вход** | **Изход** |
| -10 | Out of range |
| **Вход** | **Изход** |
| 654 | Шестстотин Педесет и Четири |
| **Вход** | **Изход** |
| 100 | Сто |
| **Вход** | **Изход** |
| 666 | Шестстотин Шестдесет и Шест |
| **Вход** | **Изход** |
| 143 | Сто Четиридесет и Три |
| **Вход** | **Изход** |
| 900 | Деветстотин |
| **Вход** | **Изход** |
| 1000 | Out of range |