**Homework: C# Advanced Topics**

This document defines the homework assignments from the [“C# HYPERLINK "http://softuni.bg/courses/csharp-basics/"BasicsHYPERLINK "http://softuni.bg/courses/csharp-basics/"“ Course @ Software University](http://softuni.bg/courses/csharp-basics/).

* **Fibonacci Numbers**

Define a method **Fib(n)** that calculates the nth [Fibonacci number](https://en.wikipedia.org/wiki/Fibonacci_number). Examples:

|  |  |
| --- | --- |
| **n** | **Fib(n)** |
| 0 | 1 |
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 5 |
| 5 | 8 |
| 6 | 13 |
| 11 | 144 |
| 25 | 121393 |

* **Prime Checker**

Write a Boolean method **IsPrime(n)** that check whether a given integer number **n** is [prime](https://en.wikipedia.org/wiki/Prime_number). Examples:

|  |  |
| --- | --- |
| **n** | **IsPrime(n)** |
| 0 | false |
| 1 | false |
| 2 | true |
| 3 | true |
| 4 | false |
| 5 | true |
| 323 | false |
| 337 | true |
| 6737626471 | true |
| 117342557809 | false |

* **Primes in Given Range**

Write a method that calculates **all prime numbers in given range** and returns them as list of integers:

|  |
| --- |
| static List<int> FindPrimesInRange(startNum, endNum)  {  …  } |

Write a method to **print a list of integers**. Write a program that enters two integer numbers (each at a separate line) and prints all primes in their range, separated by a comma.

Examples:

|  |  |
| --- | --- |
| **Start number End number** | **Output** |
| 0  10 | 2, 3, 5, 7 |
| 5  11 | 5, 7, 11 |
| 100 200 | 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199 |
| 250  950 | 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601, 607, 613, 617, 619, 631, 641, 643, 647, 653, 659, 661, 673, 677, 683, 691, 701, 709, 719, 727, 733, 739, 743, 751, 757, 761, 769, 773, 787, 797, 809, 811, 821, 823, 827, 829, 839, 853, 857, 859, 863, 877, 881, 883, 887, 907, 911, 919, 929, 937, 941, 947 |
| 100  50 | *(empty list)* |

* **Difference between Dates**

Write a program that enters two dates in format **dd.MM.yyyy** and returns the number of days between them. Examples:

|  |  |
| --- | --- |
| **First date Second date** | **Days between** |
| 17.03.2014  30.04.2014 | 44 |
| 17.03.2014  17.03.2014 | 0 |
| 14.06.1980  5.03.2014 | 12317 |
| 5.03.2014  3.03.2014 | -2 |

* **Sorting Numbers**

Write a program that reads a number **n** and a sequence of **n** integers, sorts them and prints them. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| **5**  3  -3  2  122  0 | -3  0  2  3  122 |
| **3**  0  1  0 | 0  0  1 |

* **Longest Area in Array**

Write a program to find the **longest area of equal elements** in array of strings. You first should read an integer **n** and **n** strings (each at a separate line), then find and print the longest sequence of equal elements (first its length, then its elements). If multiple sequences have the same maximal length, print the leftmost of them. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 6  hi  hi  hello  ok  ok  ok | 3  ok  ok  ok |
| 2  SoftUni  Hello | 1  SoftUni |
| 4  hi  hi  hi  hi | 4  hi  hi  hi  hi |
| 5  wow  hi  hi  ok  ok | 2  hi  hi |

* **Matrix of Palindromes**

Write a program to generate the following matrix of palindromes of **3** letters with **r** rows and **c** columns:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 6 | aaa aba aca ada aea afa  bbb bcb bdb beb bfb bgb  ccc cec cdc cfc cgc chc |
| 2 3 | aaa aba aca  bbb bcb bdb |
| 1 1 | aaa |
| 1 3 | aaa aba aca |

* **\* Longest Non-Decreasing Subsequence**

Write a program that reads a sequence of integers and finds in it the **longest non-decreasing subsequence**. In other words, you should remove a minimal number of numbers from the starting sequence, so that the resulting sequence is non-decreasing. In case of several longest non-decreasing sequences, print the leftmost of them. The input and output should consist of a single line, holding integer numbers separated by a space. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 | 1 |
| 7 3 5 8 -1 6 7 | 3 5 6 7 |
| 1 1 1 2 2 2 | 1 1 1 |
| 1 1 1 3 3 3 2 2 2 2 | 2 2 2 2 |
| 11 12 13 3 14 4 15 5 6 7 8 7 16 9 8 | 3 4 5 6 7 8 9 |

* **Remove Names**

Write a program that takes as input two lists of names and **removes from the first list all names given in the second list**. The input and output lists are given as words, separated by a space, each list at a separate line. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter Alex Maria Todor Steve Diana Steve  Todor Steve Nakov | Peter Alex Maria Diana |
| Hristo Hristo Nakov Nakov Petya  Nakov Vanessa Maria | Hristo Hristo Petya |

* **Join Lists**

Write a program that takes as input two lists of integers and **joins them**. The result should hold all numbers from the first list, and all numbers from the second list, **without repeating numbers**, and arranged in **increasing order**. The input and output lists are given as integers, separated by a space, each list at a separate line. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 20 40 10 10 30 80  25 20 40 30 10 | 10 20 25 30 40 80 |
| 5 4 3 2 1  6 3 2 | 1 2 3 4 5 6 |
| 1  1 | 1 |

* **Count of Letters**

Write a program that reads a list of letters and **prints for each letter how many times it appears in the list**. The letters should be listed in alphabetical order. Use the input and output format from the examples below. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| b b a a b | a -> 2  b -> 3 |
| h d h a a a s d f d a d j d s h a a | a -> 6  d -> 5  f -> 1  h -> 3  j -> 1  s -> 2 |

* **Count of Names**

Write a program that reads a list of names and **prints for each name how many times it appears in the list**. The names should be listed in alphabetical order. Use the input and output format from the examples below. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Peter Steve Nakov Steve Alex Nakov | Alex -> 1  Nakov -> 2  Peter -> 1  Steve -> 2 |
| Nakov Nakov Nakov SoftUni Nakov | SoftUni -> 1  Nakov -> 5 |

* **Average Load Time Calculator**

We have a report that holds dates, web site URLs and load times (in seconds) in the same format like in the examples below. Your tasks is to calculate the **average load time** for each URL. Print the URLs in the same order as they first appear in the input report. Print the output in the format given below. Use double floating-point precision. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2014-Mar-02 11:33 [http://softuni.bg](http://softuni.bg/) 8.37725  2014-Mar-02 11:34 [http://www.google.com](http://www.google.com/) 1.335  2014-Mar-03 21:03 [http://softuni.bg](http://softuni.bg/) 7.25  2014-Mar-03 22:00 [http://www.google.com](http://www.google.com/) 2.44  2014-Mar-03 22:01 [http://www.google.com](http://www.google.com/) 2.45  2014-Mar-03 22:01 [http://www.google.com](http://www.google.com/) 2.77 | [http://softuni.bg](http://softuni.bg/) -> 7.813625  [http://www.google.com](http://www.google.com/) -> 2.24875 |
| 2014-Apr-01 02:01 [http://softuni.bg](http://softuni.bg/) 8.37725  2014-Apr-01 02:05 [http://www.nakov.com](http://www.nakov.com/) 11.622  2014-Apr-01 02:06 [http://softuni.bg](http://softuni.bg/) 4.33  2014-Apr-01 02:11 [http://www.google.com](http://www.google.com/) 1.94  2014-Apr-01 02:11 [http://www.google.com](http://www.google.com/) 2.011  2014-Apr-01 02:12 [http://www.google.com](http://www.google.com/) 4.882  2014-Apr-01 02:34 [http://softuni.bg](http://softuni.bg/) 4.885  2014-Apr-01 02:36 [http://www.nakov.com](http://www.nakov.com/) 10.74  2014-Apr-01 02:36 [http://www.nakov.com](http://www.nakov.com/) 11.75  2014-Apr-01 02:38 [http://softuni.bg](http://softuni.bg/) 3.886  2014-Apr-01 02:44 [http://www.google.com](http://www.google.com/) 1.04  2014-Apr-01 02:48 [http://www.google.com](http://www.google.com/) 1.4555  2014-Apr-01 02:55 [http://www.google.com](http://www.google.com/) 1.977 | [http://softuni.bg](http://softuni.bg/) -> 5.3695625  [http://www.nakov.com](http://www.nakov.com/) -> 11.3706666666667  [http://www.google.com](http://www.google.com/) -> 2.21758333333333 |

* **Longest Word in a Text**

Write a program to find the longest word in a text. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| Welcome to the Software University. | University |
| The C# Basics course is awesome start in programming with C# and Visual Studio. | programming |

* **Extract URLs from Text**

Write a program that extracts and prints all URLs from given text. URL can be in only two formats:

* [**http://something**](http://something/), e.g. [http://softuni.bg](http://softuni.bg/), [http://forums.softuni.bg](http://forums.softuni.bg/), [http://www.nakov.com](http://www.nakov.com/)
* [**www.something.domain**](http://www.something.domain/), e.g. [www.nakov.com](http://www.nakov.com/), [www.softuni.bg](http://www.softuni.bg/), [www.google.com](http://www.google.com/)

Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| The site nakov.com can be access from [http://nakov.com](http://nakov.com/) or [www.nakov.com](http://www.nakov.com/). It has subdomains like mail.nakov.com and svetlin.nakov.com. Please check [http://blog.nakov.com](http://blog.nakov.com/) for more information. | [http://nakov.com](http://nakov.com/)  [www.nakov.com](http://www.nakov.com/)  [http://blog.nakov.com](http://blog.nakov.com/) |

* **\* Counting a Word in a Text**

Write a program that counts **how many times a given word occurs in given text**. The first line in the input holds the word. The second line of the input holds the text. The output should be a single integer number – the number of word occurrences. Matching should be **case-insensitive**. Note that **not all matching substrings are words** and should be counted. A **word** is a sequence of letters separated by punctuation or start / end of text. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| hi  **Hi**dden networks say “**Hi**” only to **Hi**tachi devices. **Hi**, said Matu**hi**. **HI**! | 3 |
| SoftUni  The Software University (**SoftUni**) trains software engineers, gives them a profession and a job. Visit us at [http://**HYPERLINK "http://softuni.bg/"softuni**HYPERLINK "http://softuni.bg/".bg](http://softuni.bg/). Enjoy the SoftUnification at **SoftUni**.BG. Contact us.Email: [INFO@**HYPERLINK "mailto:INFO@SOFTUNI.BG"SOFTUNI**HYPERLINK "mailto:INFO@SOFTUNI.BG".BG](mailto:INFO@SOFTUNI.BG). Facebook: <https://www.facebook.com/SoftwareUniversity>. YouTube: <http://www.youtube.com/SoftwareUniversity>. Google+: <https://plus.google.com/+SoftuniBg/>. Twitter: <https://twitter.com/softunibg>. GitHub: [https://github.com/**HYPERLINK "https://github.com/softuni"softuni**](https://github.com/softuni) | 5 |
| SoftUni  Software University | 0 |
| SoftUni  **SoftUni** | 1 |
| SoftUni  **SoftUni**.**SoftUni** | 2 |

* **\* Perimeter and Area of Polygon**

Write a program that **calculates the perimeter and the area of given polygon** (not necessarily convex) consisting of **n** floating-point coordinates in the 2D plane. Print the result rounded to two decimal digits after the decimal point. Use the input and output format from the examples. To hold the points, define a class **Point(x,** **y)**. To hold the polygon use a **Polygon** class which holds a **list of points**. Find in Internet how to calculate the [polygon perimeter](http://www.mathopenref.com/polygonperimeter.html) and how to calculate the [area of a polygon](http://www.mathopenref.com/coordpolygonarea.html). Examples:

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 3  0 0  0 1  1 1 | perimeter = 3.41  area = 0.50 |  |
| 7  -2 1  1 3  5 1  1 2  1 1  3 -2  -2 1 | perimeter = 22.64  area = 9.5 |  |