# Homework: C# Advanced Topics

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

## 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 |

## 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://forums.softuni.bg>, <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> 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> for more information. | <http://nakov.com>  [www.nakov.com](http://www.nakov.com)  <http://blog.nakov.com> |

# Exam problems.\*\*

**All of the problems below are given from Variant 8 of C# Basics Practical Exam (30 August 2015). You are not obligated** to submit any of them in your homework. We highly recommend you to try solving some or all of them so you can be well prepared for the upcoming exam. You need to learn how to use conditional statements, loops, arrays and other things (learn in internet how or read those chapters in the book “[Fundamentals of computer programming with C#](http://www.introprogramming.info/intro-csharp-book/read-online/)”). If you still find those problems too hard for solving it’s very useful to **check** and **understand** the solutions. You can download all solutions and tests for this variant [here](http://svn.softuni.org/admin/svn/csharp-basics/June-2015/Programming-Basics-Exam-30-Aug-2015.zip) or check all [previous exams](https://softuni.bg/courses/programming-basics). You can also test your solutions in our automated [judge system](https://judge.softuni.bg/Contests/104/Programming-Basics-Exam-30-August-2015) to see if you pass all tests.

## \* Daily Calorie Intake

Kalinko is a junior software developer, who is mostly doing freelance work for random contractors. He has just been hired by a fitness instructor and is been tasked to create a software, which will assist the instructor in preparing healthy diets for his customers. In order to prepare the diets, the fitness instructor needs to know the necessary daily calorie intake for each of his clients. The daily calorie intake (DCI), is calculated by **multiplying the Basal Metabolic Rate (BMR) of the client, by a constant, which is determined by the number of workouts that the person does per week**. The BMR is calculated with the following formula:

**Men: BMR = 66.5 + (13.75 x weight in kg) + (5.003 x height in cm) – (6.755 x age in years)**

**Women: BMR = 655 + (9.563 x weight in kg) + (1.850 x height in cm) – (4.676 x age in years)**

Once the BMR is calculated, we can get the person's DCI, using the following table:

|  |  |
| --- | --- |
| **Number of workouts** | **Daily Calorie Intake** |
| No workouts | DCI = BMR \* 1.2 |
| 1–3 workouts per week | DCI = BMR \* 1.375 |
| 4–6 workouts per week | DCI = BMR \* 1.55 |
| 7–9 workouts per week | DCI = BMR \* 1.725 |
| Extra heavy workouts | DCI = BMR \* 1.9 |

Also, the fitness instructor that hired Kalinko, lives in the United States, which means that the weight and the height of his clients, will be given in an **Imperial format – pounds (lbs.) for the weight and inches for the height**. In order to make the BMR formulas work, Kalinko will have to **convert Imperial values into Metric values**. Assume that **1 inch has 2.54cm and 1kg has 2.2lbs**.

Your job is to help Kalinko with his first big contract and create the software for him. You will be given a person's weight in pounds (lbs.), height in inches, age, gender and number of weekly workouts, each at a separate line. Your only output, should be the person's daily calorie intake (DCI). The DCI should be **rounded down to the nearest integer number.**

### Input

The input should be read from the console. It will consist of exactly 5 input values, each at a separate line.

1. **W – weight in pounds (lbs.)**
2. **H – height in inches**
3. **A – age**
4. **G – gender**
5. **E – workouts per week**

### Output

* The output should be the calculated DCI. It should be a **single number, rounded down to the nearest integer number.**

### Constraints

* The W, H, A and E inputs will be valid integers, in the range [-2,147,483,648 … 2,147,483,647]
* The G input will be a single character – 'm' for male or 'f' for female
* Allowed working time for your program: 0.25 seconds.
* Allowed memory: 16MB.

### Examples

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | **Comments** | **Input** | **Output** | **Input** | **Output** |
| 154  70  27  m  0 | 2083 | 70 inches \* 2.54 = 177.8cm;  154lbs / 2.2 = 70kg;  66.5 + (13.75 \* 70) + (5.003 \* 177.8) – (6.755 \* 27) = 1736.1484 \* 1.2 = 2083.37808 | 130  63  21  f  7 | 2445 | 250  85  35  m  15 | 4698 |

## \* The football statistician

Mr. Vulchan is a football statistician and his daily job is to watch football matches, collect the results and prepare a table of the league. Unfortunately, one day he fell off a cliff and hit his head badly. From that day he can’t remember any numbers and he needs some help with them. Your task is to prepare the league table for him**.** Since you are very good at programming, you decide to make a program to do this.

You should make a league table only for one of the leagues. It consists of 8 teams: Arsenal, Chelsea, Manchester City, Manchester United, Liverpool, Everton, Southampton and Tottenham.

You are given a string in the format **″team1** outcome **team2″** separated by **one or more whitespaces**. The **outcome** of the match will be one of the characters **[1, X, 2].** The character **′1′** represents a **win** for **team1**, the character **′2′** represents a **win** for **team2** and **′X′** represents **draw**.

* When **one of the teams** wins, it receives **3 points.** The **other team** receives **0 points**.
* In case of a **draw** the both teams receive **1 point**.

Mr. Vulchan will pay you **N** **euros** for every match. You should evaluate how much money you will obtain for this job and print it on the console in **leva**. Assume that **1** euro is **1.94lv**.

### Input

The input data should be read from the console.

* On the first line you will receive the payment for every match.
* On the next **N** lines you are given several input lines holding the match and its result. When you receive the command “**End of the league.**” the program should stop.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

The output data should be printed on the console.

* On the first line you should print the evaluated price for all matches in the league in **leva** rounded to **two** digits after the decimal point.
* On the next 8 lines you should print all the teams in **alphabetical order**, each on a separate line, together with the **points**, they have gained.

### Constraints

* The payment will be a floating point number in the range (-7.9 x 1028 to 7.9 x 1028) / (100 to 1028)
* Allowed working time for your program: 0.1 seconds.
* Allowed memory: 16 MB.
* Hint: The teams with more than one word are represented without **spacing** in **Pascal case.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 24.33  Chelsea 2 ManchesterCity  Everton 1 ManchesterUnited  Arsenal X Liverpool  Southampton 1 Tottenham  End of the league. | 188.80lv.  Arsenal - 1 points.  Chelsea - 0 points.  Everton - 3 points.  Liverpool - 1 points.  Manchester City - 3 points.  Manchester United - 0 points.  Southampton - 3 points.  Tottenham - 0 points. |

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12.33  Chelsea 1 ManchesterUnited  Everton 1 ManchesterCity  Arsenal 1 Tottenham  Southampton 2 Liverpool  Southampton X ManchesterCity  Liverpool X Everton  Chelsea 2 Arsenal  ManchesterUnited X Tottenham  End of the league. | 191.36lv.  Arsenal - 6 points.  Chelsea - 3 points.  Everton - 4 points.  Liverpool - 4 points.  Manchester City - 1 points.  Manchester United - 1 points.  Southampton - 1 points.  Tottenham - 1 points. |

## \*\* Striped Towel

The summer is almost over, but you still plan to have a summer vacation on the beach. Since our Black Sea is quite polluted nowadays, you plan to go to the Mediterranean Sea. A lot of people are going there, so you want to stand out of the crowd on the beach. You want to create your own design for a beach towel and have it in any possible size, so you can order to a very well-known local producer a full package for your family.

Your first steps in programming have led you to the conclusion that you can create a program to display your design. The width of the towel will be given as integer number. The height is calculated as the width, multiplied by 1,5 and rounded down to the nearest integer. The design, you have chosen, is one with diagonal stripes. It follows this pattern:

The diagonal stripes are represented by **′#′.** The space between the stripes is represented as **′..′.** The towel always starts with a stripe in the top-left corner and follows the pattern stripe-space-stripe-space. The stripes are going from top-right to bottom-left. See the examples for more clarity.

### Input

The input should be read from the console. It will consist of a single line, holding the width of the towel.

### Output

The output should be the towel design, based on the input value.

### Constraints

* The width will be valid integer, in the range [3 … 151].
* Allowed working time for your program: 0.25 seconds.
* Allowed memory: 16MB.

### Examples

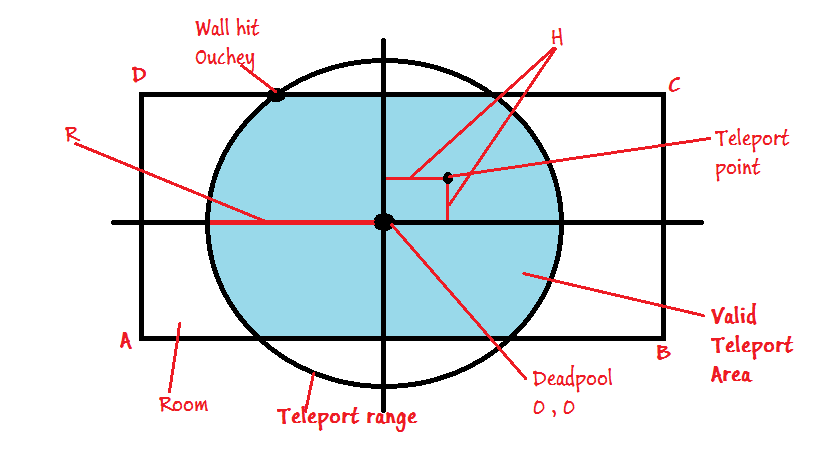
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 7 | #..#..#  ..#..#.  .#..#..  #..#..#  ..#..#.  .#..#..  #..#..#  ..#..#.  .#..#..  #..#..# |  | 9 | #..#..#..  ..#..#..#  .#..#..#.  #..#..#..  ..#..#..#  .#..#..#.  #..#..#..  ..#..#..#  .#..#..#.  #..#..#..  ..#..#..#  .#..#..#.  #..#..#.. |  | 5 | #..#.  ..#..  .#..#  #..#.  ..#..  .#..#  #..#. |

## \*\* Teleport Points

# Are you familiar with the Marvel universe? Chances are that you have at least heard of some of Marvel's most prominent heroes, such as Spider-Man, Captain America, Iron Man and Wolverine. Chances are, however that you haven't heard about some of Marvel's anti-heroes, mostly because anti-heroes tend to kill people, a LOT of people and thus don't usually get endorsed by the family-friendly mindset of Hollywood. Among these anti-heroes, is a character named Wade Wilson, also known as Deadpool, or “the merc with the mouth”. Wade is best known for his total disregard for human life, for being insane, for being practically immortal, and for having a love affair with Death herself. Deadpool's gear primarily consists of twin katana swords and two automated pistols. However, in some instances, he also possesses a teleportation device, which allows him to “body-slide” to short distances. The device has been destroyed during his last mission and he is now trying to acquire a new one. He already has the device built and he just needs someone to create the software for it, which will calculate possible teleportation points within a certain radius. Since you are the first computer programmer that he finds, you are appointed to do the job. You wouldn't want to disappoint Deadpool now, wouldn't you?

The program is very basic. The device can **teleport** a person **to any point** within a **radius R**. It also has a **step H**, which is the **distance between two neighboring points**. H has to be set manually, so **it cannot change** in the middle of a calculation. The device can only be used **within rectangular rooms** and the teleport **cannot pass through walls**. Deadpool knows that you are a junior developer, so your algorithm must work only **within a two-dimensional space**.

You will be given the coordinates of the **four points of the rectangle**, which will represent **the room**. You will also be given the **value R**, which is the **radius** of the device and the **value H**, which is the device's **step**. Deadpool's position will **always be 0,0**. Your task is to count **all possible points** Deadpool can **teleport** **to**. Points that will make him teleport **within a wall**, are **not considered valid**.

You should probably start working now. Deadpool does not regard patience as a virtue. He did draw you a pretty picture however, so that you can get a better understanding of your task.

### Input

The input should be read from the console and will consist of exactly 6 lines.

* The first **4 lines** will contain the **[X Y] coordinates** for each of the **four points** of the room.
* The **X** and the **Y** values will be separated by a **single space**.
* **Lines 5 and 6** will contain the values of the radius **R** and the step **H** respectfully.

1. **[X Y]** – coordinates for point A
2. **[X Y]** – coordinates for point B
3. **[X Y]** – coordinates for point C
4. **[X Y]** – coordinates for point D
5. **R** – Radius
6. **H** – Step

### Output

* The output should consist of a single number, representing the count of valid teleport points.

### Constraints

* **X and Y** will always be in the range [-30.5 … 30.5]
* **R and H** will always be in the range [0.1 … 30.5]
* **A, B, C, D** will always form a rectangle.
* Allowed working time for your program: 0.25 seconds.
* Allowed memory: 16MB.

### Examples

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | **Explanation** |  | **Input** | **Output** |
| -20 -3  20 -3  20 3  -20 3  10  0.4 | 731 | The room is **40** wide and **6** tall. The circle has radius of **10**. Total points within the area with step **0.4** are **731**. |  | -7.2 -8.8  10.4 -8.8  10.4 9.1  -7.2 9.1  30.5  1 | 324 |