

HW 5

Skills needed: loops, if sentences, static methods, string manipulation.

String examples: <https://www.dotnetperls.com/string-char>, <https://www.w3resource.com/csharp-exercises/string/index.php>, <https://www.dotnetperls.com/substring>

You do not need to create classes; you can do all the tasks as static methods. Before starting a task think it through! Use a piece of paper and write some example values!

- 1) Compound interest (*liitintress*) is made up from the initial amount of money and the yearly interest. After every year the final sum is calculated from interest rate and initial amount of money.

For example: if initial amount is 2200 and interest rate is 2.1%:

*1st year: interest is $2200 * 0.021 = 46.2$; total amount is 2246.2 (2200+46.2)*

*2nd year: interest is $2246.2 * 0.021 = 47.17$; total amount is $2246.2 + 47.17 = 2293.3702$*

..

Create the method that asks the initial amount of money, interest rate and years for keeping the money and based on that calculates the final sum.

Example:

FindCompoundInterest(2000, 0.02, 2) - > 3216.87

- 2) We kept 6 important id codes in a .txt file. At night someone made a prank and cut the id codes in half and put them into two separate files. All id codes were cut from a different position and we know that the total length of the id code has to be 11.

For example:

if first part of the file is with length 3, we know that the other part has to be with length 8 (11-3=8). Based on this try to restore the id codes back together. Create a new file where id codes are stored.

Beginnings of the id code are in the first column, the endings are in a second column. Take codes from the first file and find matching endings (total length of the code is always 11!) from the second file. Each beginning has only one matching ending.

Copy file contents and create corresponding text files in your computer.

| Contents of the first file (beginnings of the id codes) | Contents of the second file (endings of the id codes) |
|---|---|
| 378100151 | 810015233 |
| 3781001 | 79 |
| 37 | 015222 |
| 37810 | 0015190 |
| 3 | 7810015233 |
| 3781 | 5190 |

Example:

Beginning: 378100151; Ending: 79; full id code: 37810015179

Beginning length is 9; ending has to be with length 2.

Help: Use lists to store values from both file (we need to read in beginnings and endings and add them to different lists. Use 2 separate StreamReaders() for reading in the values.)

Add a third list where the results (full id codes) will be kept. Use for loops and the knowledge that the length of the code has to be 11 to put the parts (ending, beginning) together.

string idCode = beginning+ending

- 3) Write a method for finding a median (*mediaan*) and arithmetic mean (*aritmeetiline keskmine*) from a list of integers.

Median is the middle value of the sorted list. If there are 2 middle values (even numbers of items in a list) the median is two middle numbers summed up and then divided by 2.

Arithmetic mean is sum of all items divided by the amount of items.

For example:

{3, 3, 5, 9, 11} -> median is 5. Arithmetic mean is 6.2

{3, 5, 7, 9} -> median is 6; $(5 + 7) / 2 = 6$. Arithmetic mean is 6.

You can create the lists yourself inside that method (no parameters) or use the list as a parameter. Use different values for testing purposes.

- 4) We are hired by a bank and need to write a piece of code that outputs the information about money. User can enter an amount of money (max 9 euros and 99 cents) in numbers and we have to output this information as text. NB! Grammatical correctness is important!

Example:

GetMoneyInfo("207")-> "2 eurot ja 7 senti"

GetMoneyInfo ("101")-> "1 euro ja 1 sent" (NB! mitte "1 eurot ja 1 senti", ainsusel ja mitmusel tuleb vahet teha)

GetMoneyInfo ("95")-> "95 senti" (st. ilma eurodeta)

GetMoneyInfo ("100")-> "1 euro" (st. ilma sentideta)

- 5) Write code that prints out info about peoples salaries. In order to make calculations easier lets assume that net salary (*netopalk*) is: gross salary (*brutopalk*) minus 25%.

*Example: gross salary is 2000; net salary is then $2000 - (2000 * 0.25) = 1500$*

In input file net salary starts with letter "N" and gross salary with "G". In the output file we just keep numbers.

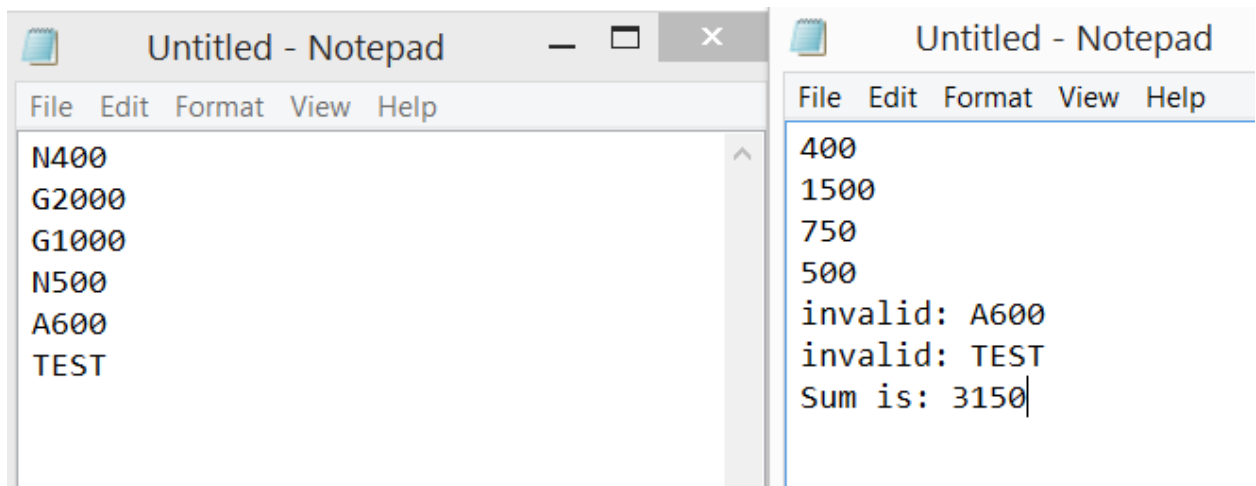
We have an input file where net and gross salaries are mixed. We want to get a result file which:

- Contains only net salary values (if input is gross salary then calculate net salary)
- Contains the sum of all net salaries in the end of the file

If input value is:

- Gross salary -> we calculate net salary and write it to results file
- Net salary -> we just write the value to results file
- invalid value (does not start with 'N' or 'G' and is not followed by numbers) is presented then we write "invalid: + value" to results file.

Example: input on the left and output on the right.



```
File Edit Format View Help
N400
G2000
G1000
N500
A600
TEST
```

```
File Edit Format View Help
400
1500
750
500
invalid: A600
invalid: TEST
Sum is: 3150|
```

Help: use lists. Create a separate method for calculating the salary. Do it step by step! Add comments for yourself. Think of the order of doing things.