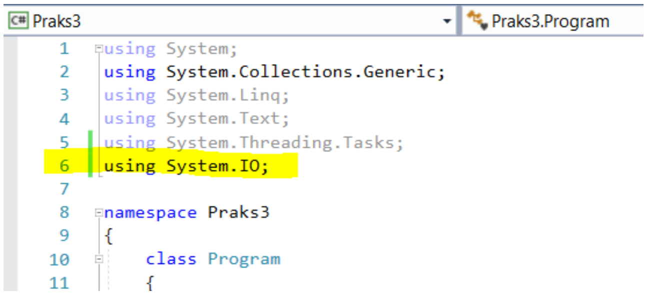
NB! We need to use methods from other assemblies, so we need to add using sentence; using System.IO;



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;

namespace Praks3
{
    class Program
    {
```

1) **File path. Read and look at the example**: https://www.dotnetperls.com/path
   a. Create a text file called “test.txt”, save it to disk and add some content (for example four questions on separate lines)
   b. Print out file full path, extension and filename.

2) Reading and writing to text file

StreamReader is used for reading file content, StreamWriter for writing to text file.

They are wrapped in using statement, which shows that the objects will be disposed after their usage.

```
  string filePath =
    "C:\\Users\\liisa\\source\\test.txt"; //has to contain text
  string writingFilePath =
    "C:\\Users\\liisa\\source\\fileToWriteAt.txt"; //does not have to exist
string line;
//reading from a file
using (StreamReader reader = new StreamReader(filePath))
{
    while ((line = reader.ReadLine()) != null)
    {
        Console.WriteLine(line);
        Console.WriteLine("----------");
    }
} //reader is disposed


//appending text to a file
using (StreamWriter writer = new StreamWriter(writingFilePath, true))
{
    writer.WriteLine("some text");
}//reader is disposed

//overwriting text in a file
using (StreamWriter writer = new StreamWriter(writingFilePath, false))
{
    writer.WriteLine("overwrite text");
}//reader is disposed
```

StreamReader(). Read: https://www.dotnetperls.com/streamreader

a. Create method called ReadTextByLine(parameter fileLocation) which prints out all the text line by line from a file

StreamWriter(). Read: https://www.dotnetperls.com/streamwriter

b. Create a method for adding text to file (overwrites existing text); text is added as string parameter
c. Create a method for adding text to file (appends to text, does not overwrite); text is added as string parameter (https://msdn.microsoft.com/en-us/library/36b035cb(v=vs.110).aspx)

DIY Task 1:

Write a method for sorting values from a file and save the result to another file.

1) Read all values from text file A (it should contain multiple lines) and save them to a list
2) Sort the list (so that values are alphabetically ordered)
3) Write the sorted list to text file B

DIY Task 2:

Create a method that asks 3 questions from file A (questions.txt), asks question from user and then saves answer to another file B (answers.txt). Different answers from different users are stored in different sections.

Append answers to the file, do not overwrite them

Possible scenario:

Please enter your name

-Read user name and save to file.

Ask question 1 from file

-Read answer to question 1 and save it (with question number)

Ask question 2 from file

-Read answer to question 2

***Advanced**:*

3) **Reading from a .csv file** and storing values in a List<string[]>

In csv (CommaSeparatedValue) file different values are kept together and separated by ';' sign. Data structure is fixed and each column has its meaning. In our case: Name; Age; Income.

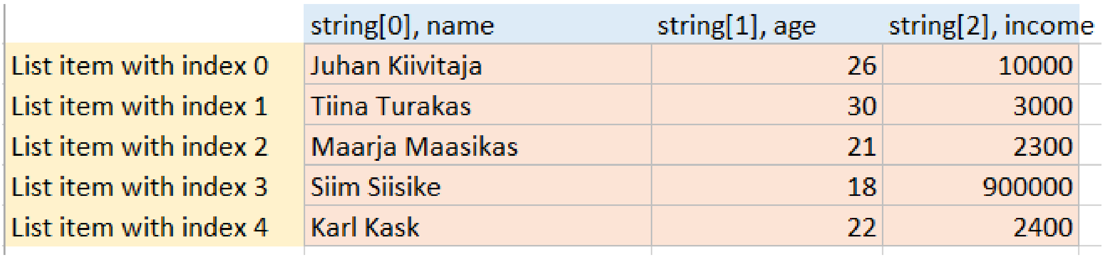
Copy this to be the content of your file:

Juhan Kiivitaja;26;10000
Tiina Turakas;30;3000
Maarja Maasikas;21;2300
Siim Siisike;18;900000
Karl Kask;22;2400

If we split the line at ‘;’ then the resulting array of first line would be:

array[0] = “Juhan Kiivitaja”, array[1] = “26”, array[2] =”10000”. So we know that name is always in column with index 0, age with index 1 and income with index 2.

After reading in the data, we want to store it in a list containing arrays. Structure would be:



	string[0], name	string[1], age	string[2], income
List item with index 0	Juhan Kiivitaja	26	10000
List item with index 1	Tiina Turakas	30	3000
List item with index 2	Maarja Maasikas	21	2300
List item with index 3	Siim Siisike	18	900000
List item with index 4	Karl Kask	22	2400

Code for reading in the file:

```
using (var reader = new StreamReader("test.csv"))
     {
         List<string[]> listA = new List<string[]>(); //we create a new list
         while (!reader.EndOfStream) //until the file is not ended
         {
             string line = reader.ReadLine(); //we read the line
             var values = line.Split(';'); //we read the line and split it by ';'
creating an array

             listA.Add(values); //we add the resulting array to list

         }
}
```

Read before: https://www.tutorialspoint.com/csharp/switch_statement_in_csharp.htm

DIY Task 2: we have to give out money to young people based on their age and income. The rules are as follows:

a) If age is 18 or 19 we give out 400 units of money.
b) If age is 20 -25 we give income divided by 10 (If income was 100, we give out 10)
c) If age is 26 -29 we give out the same amount as income but not more than 8000.
d) If age is 30 or more we give out no money.

Print out name and the amount of money this person would receive.