

1 Intro session

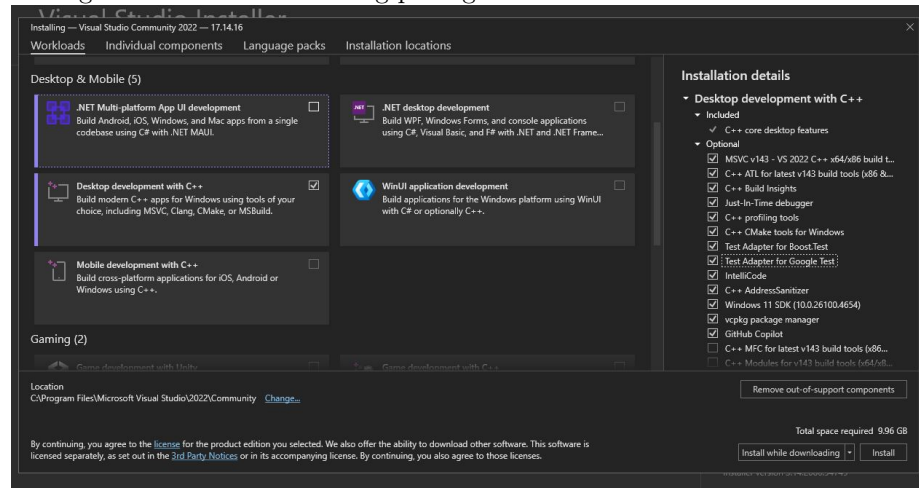
To begin with, make sure you have read the Laboratory Guide on Moodle. In this lab, you will learn how to write a C/C++ program in Microsoft Visual Studio or JetBrains CLion. You will also learn how to generate data for evaluating algorithms and how to create graphs in Microsoft Office Excel.

1.1 Microsoft Visual Studio

1.1.1 Install

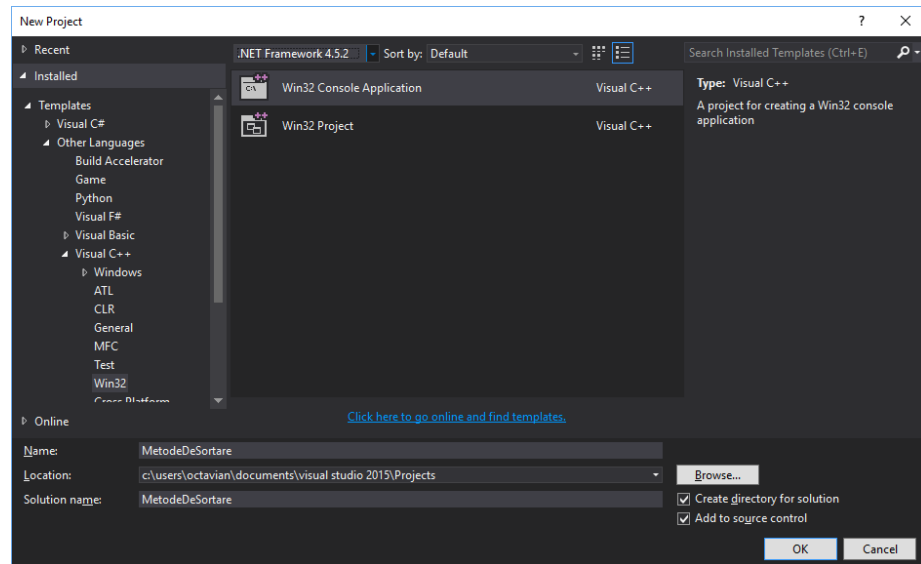
Free edition: <https://visualstudio.microsoft.com/vs/community/>

During installation the following packages must be chosen:

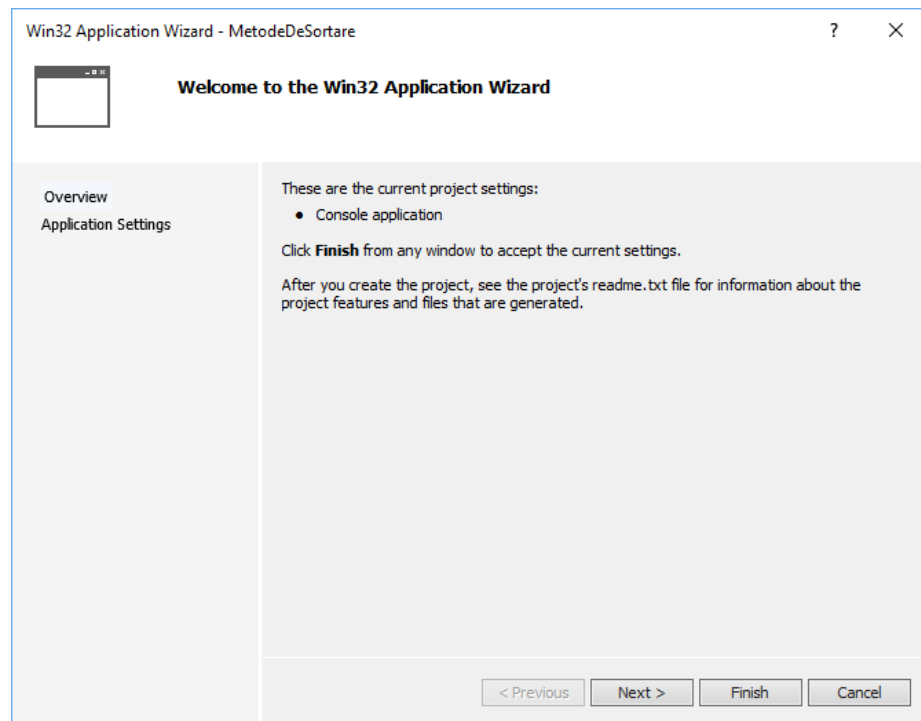


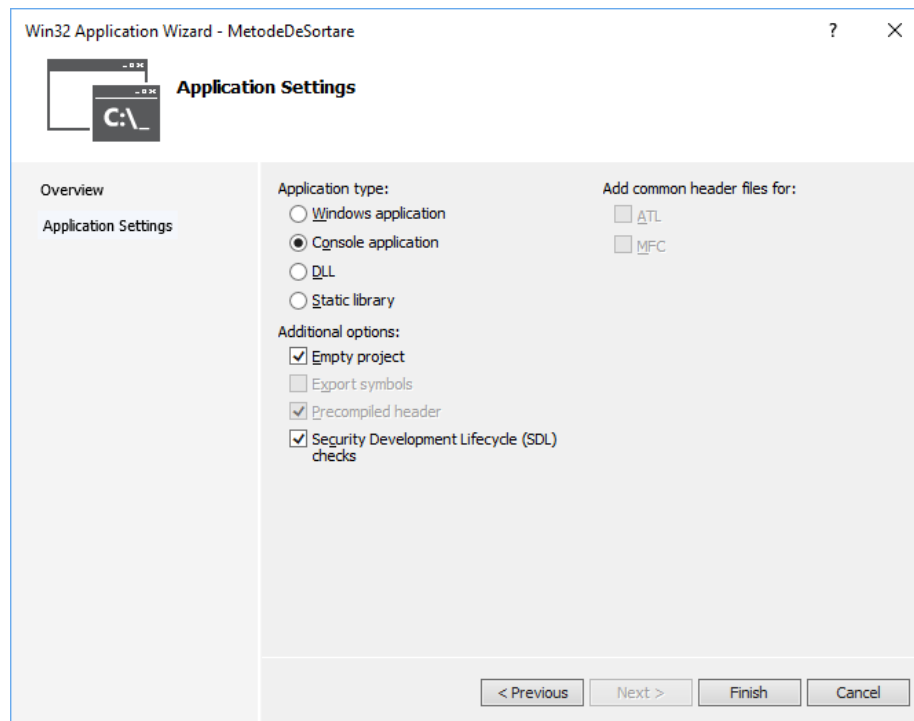
1.1.2 Use

1. Create project: *File -> New -> Project...*
2. Project type: *Templates -> Other Languages -> Visual C++ -> Win32 -> Win32 Console App*

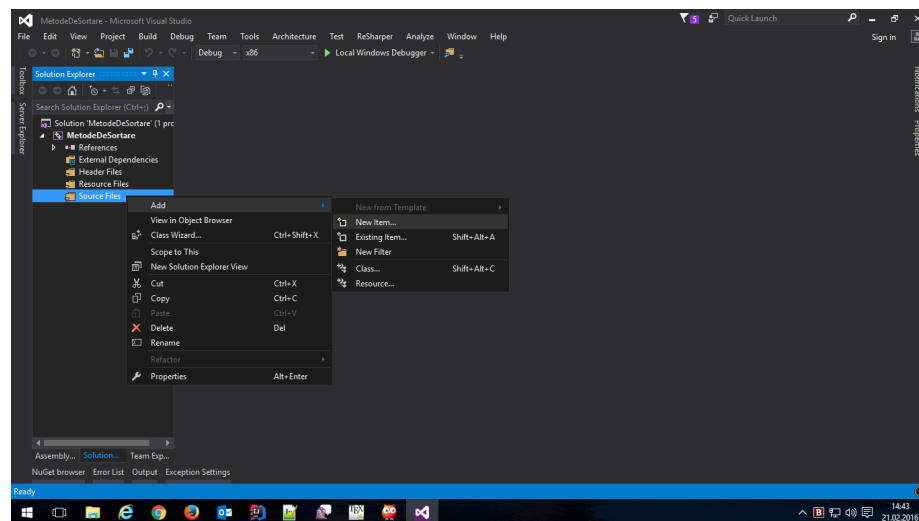


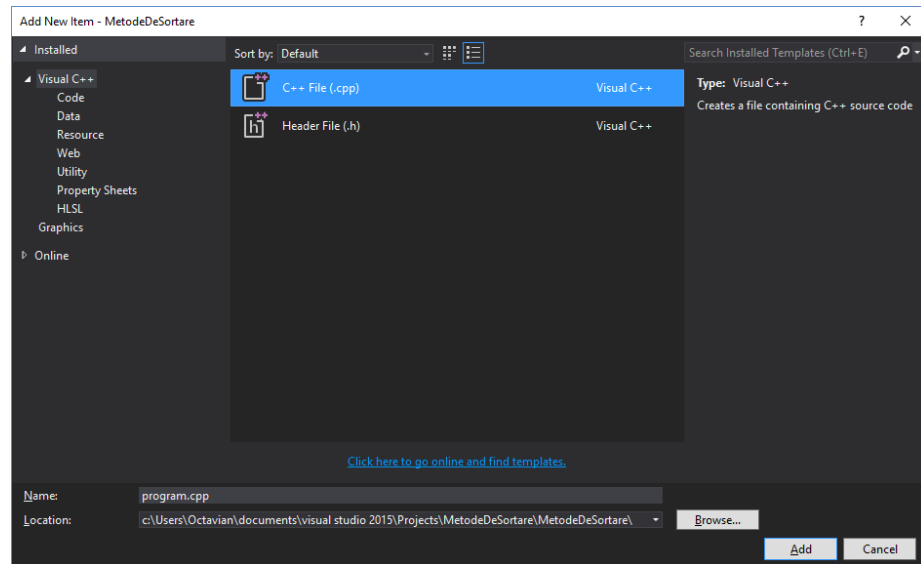
3. Project properties: choose “*Console application*” and select “*Empty project*”



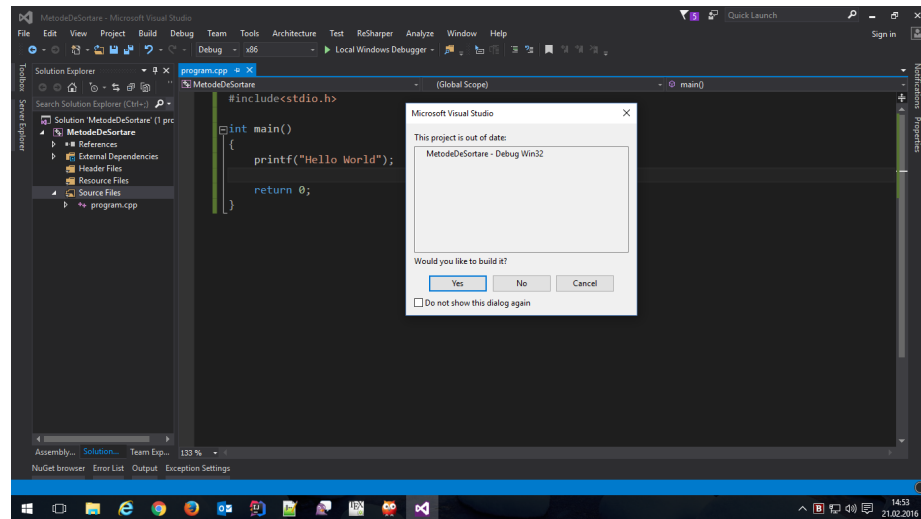


4. Create a *.cpp file





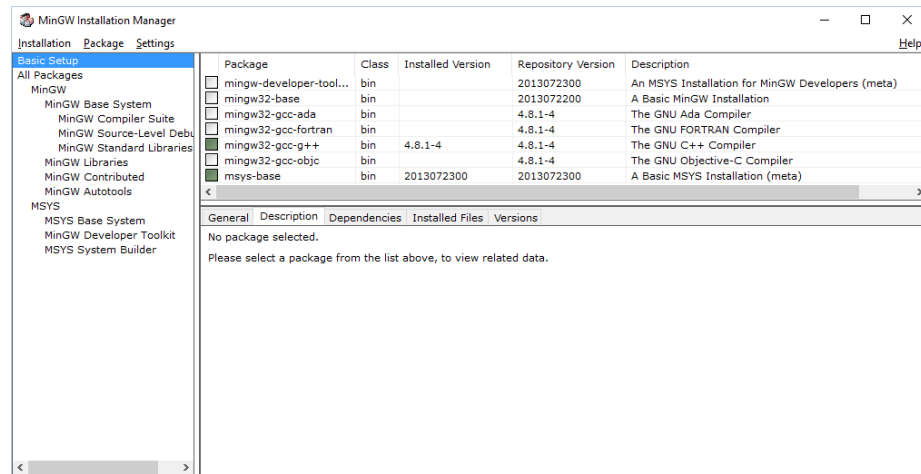
5. Compile and execute the program (in **DEBUG** mode)



1.2 JetBrains CLion

1.2.1 Install

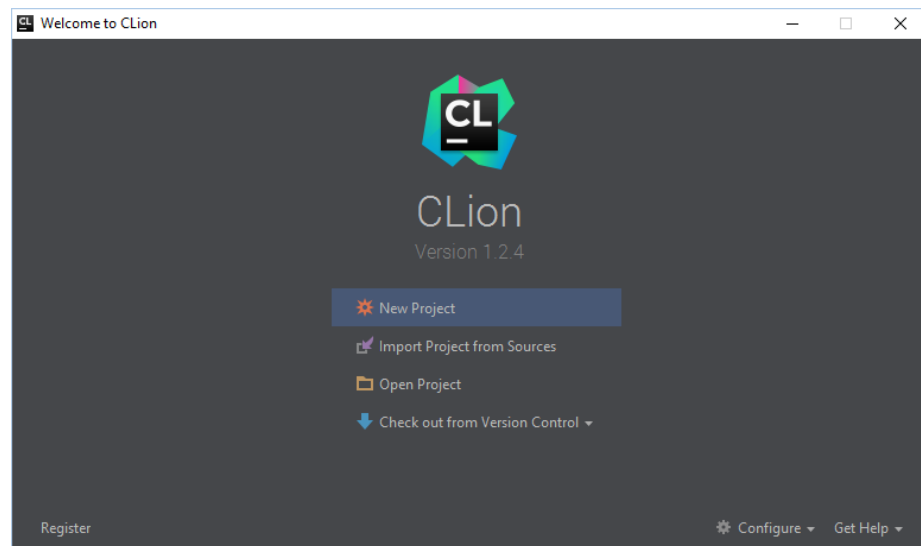
Register with your student e-mail address: *@student.utcluj.ro* on <https://www.jetbrains.com/student/>
 Download and install MinGW: <https://sourceforge.net/projects/mingw/>



Download and install CLion: <https://www.jetbrains.com/clion/download/#section=windows>

1.2.2 Use

1. Create project: *New Project*



2. Wait until symbols are loaded
3. Create a *.cpp file

1.3 C/C++

1.3.1 Reading/writing files

Exercise - steps:

- Declare an array v of length MAX_SIZE (a constant defined by you)
- Read an n from the keyboard
- Open the *input.txt* file, read n numbers from it and save them in array v
- Save the n numbers in the *output.txt* file in **reverse** order

1.3.2 Generating test cases

To test the algorithms you will implement, you will need to use a series of input data: sorted ascending arrays, sorted descending arrays, random arrays, etc. Generating ascending/descending arrays should be straightforward. For generating random arrays, you can use the following:

- *Profiler* library from Moodle (or <https://github.com/cyprioprism/utcn-fa-profiler>)
- *rand()*, *srand()* methods, read:
 - <http://www.cplusplus.com/reference/cstdlib/rand/>
 - <http://www.cplusplus.com/reference/cstdlib/srand/>
 - http://www.cplusplus.com/reference/cstdlib/RAND_MAX/

Exercise - steps:

- Read n , min and max from the keyboard
- Generate a random array of n elements with values bounded within min and max
- The array must be different for each execution of the program
- Add the array to the *output.txt* file

1.3.3 Generating plots

For the generation of the charts, you can use:

- *Profiler* library from Moodle (or <https://github.com/cyprioprism/utcn-fa-profiler>)
- Microsoft Office Excel

1.3.4 Microsoft Office Excel

You will need to create a file with the *.csv* (comma-separated values) extension. The file should have a structure similar to the following:

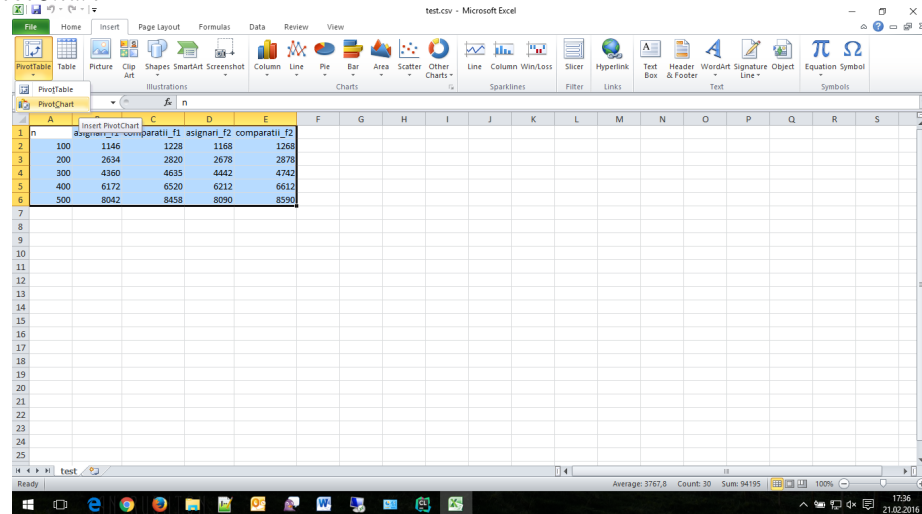
```
n,assignments_f1,comparisons_f1,assignments_f2,comparisons_f2
100,1146,1228,1168,1268
200,2634,2820,2678,2878
300,4360,4635,4442,4742
400,6172,6520,6212,6612
500,8042,8458,8090,8590
```

Legend:

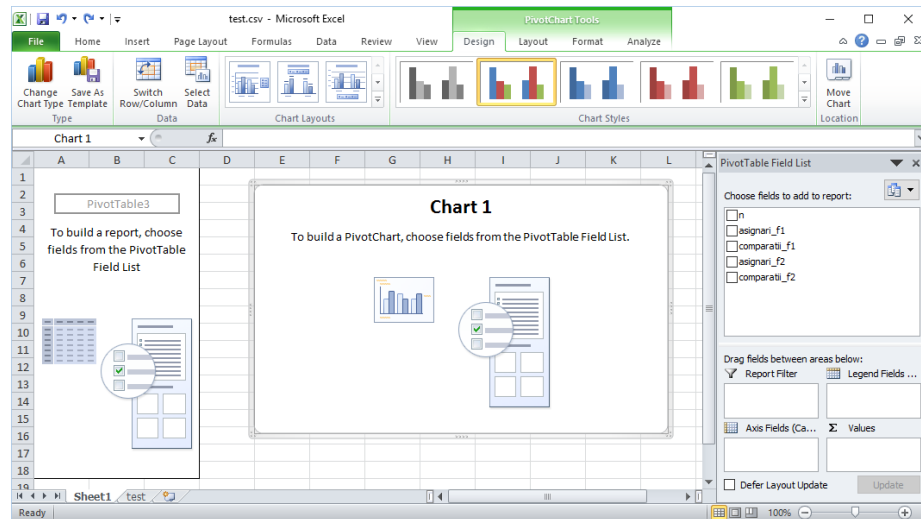
- n = the size of the problem (ex: lungimea şirului de intrare)
- assignments_f1= the number of assignments for the best-case scenario of method 1
- comparisons_f2= the number of assignments for the best-case scenario of method 2

Caution: If you open the CSV file in Excel and the values appear in a single column, it means you should use a different delimiter character (e.g., use a semicolon ”;”).

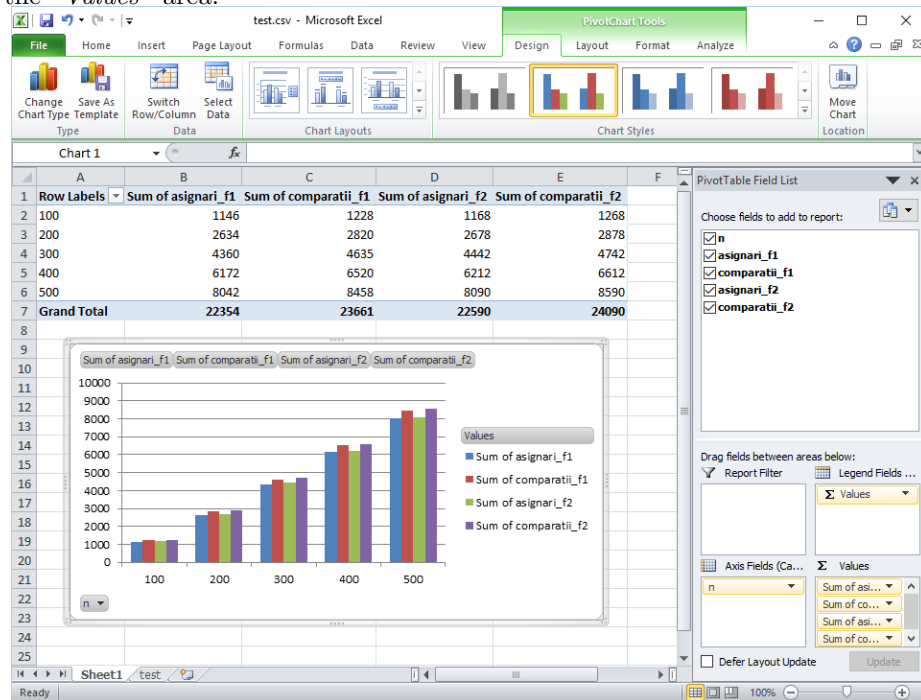
After you've opened the CSV file in Excel, select all the values and create a *PivotChart*.



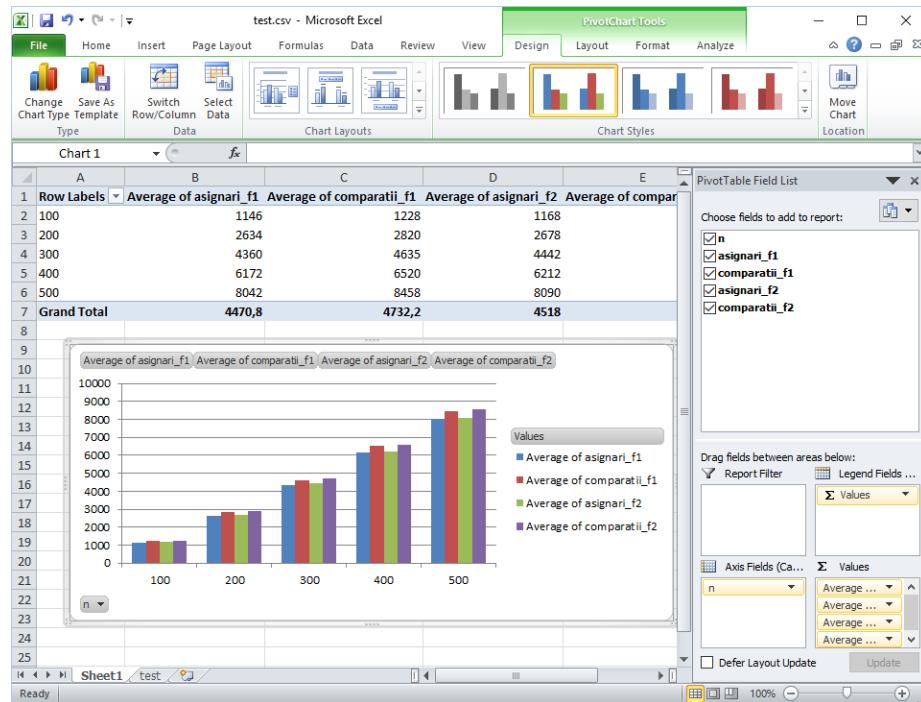
After you click "Ok," the window should look like the picture below.



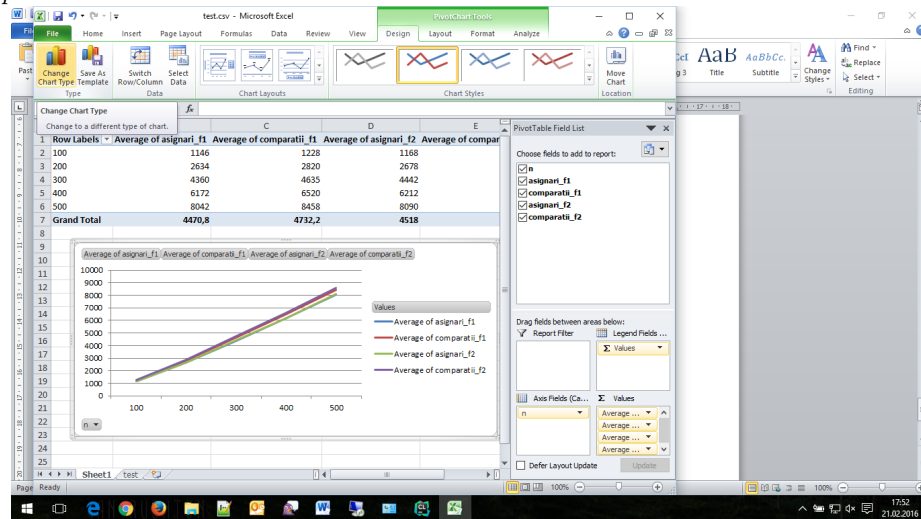
In the left panel, drag "n" to the "Axis Fields" area and the other columns to the "Values" area.



Change the aggregation function from "Sum" to "Average": Click on the black arrow in each row in the "Values" area, then choose "Value Field Settings" and select "Average." If you've changed them correctly, the window should look like the picture below.



The final step is to change the chart type to a line chart from "Change Chart Type." The final result should look like this:



1.3.5 Exercise

Write a program that for each value n of the interval $\{100, 200, \dots, 10.000\}$ computes and adds into a file the following values:

$n, 100 \cdot \log(n), 10 \cdot n, n \cdot \log(n), 0.1 \cdot n^2, 0.01 \cdot n^3$

Use the values from the files to generate a chart depending on n .