

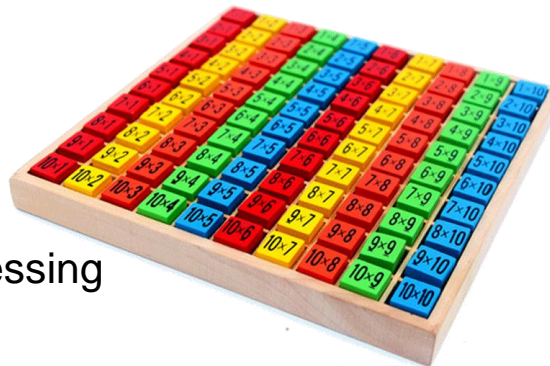
## LAB 8: DATA PROCESSING (FILE AND STRING)

BME 121 2016

Rasoul Nasiri

# Topics

- File and string processing
- CSV file and string processing
- Shakespear poems and words processing
- WA5
- Cryptography



# File I/O review the commands

- 1 `using System.IO;`
- 2 `FileStream inFile = new FileStream(@"myfile.txt", FileMode.Open, FileAccess.Read);`
- 3 `StreamReader inStream = new StreamReader(inFile);`
- 4 `string line = inStream.ReadLine();`
- 5 `inStream.Dispose();  
inFile.Dispose();`

# Review string processing

```
string a = "cool";
string[ ] b = {"1", "co"};
```

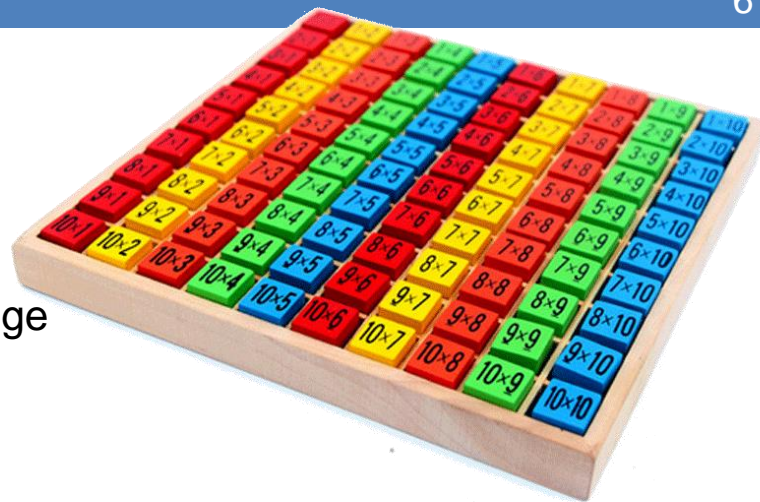
Method	What does it do?	Example	Result
Split(char[] delimiters)	Splits the string into an array of strings at every delimiter character	(see previous slides)	
Join(string delimiter, string[] arr)	Opposite effect as Split	String.Join("  ", b);	"1  co"
StartsWith(string x)	Sees if the string starts with the same characters as string x	a.StartsWith("co"); a.StartsWith("k");	True False
EndsWith(string y)	Sees if the string ends with the same characters as string y	a.EndsWith("ol"); a.EndsWith("cc");	True False
Contains(string z)	Sees if string z occurs within the string	a.Contains("oo"); a.Contains("cl");	True False
IndexOf(string x)	Returns the index of the first occurrence of string x, or -1 if it can't find it	a.IndexOf("o");	1
LastIndexOf(string x)	Returns the index of the last occurrence of string x, or -1 if it can't find it	a.LastIndexOf("o");	2

# Review string processing

Method	What does it do?	Example	Result
<code>Insert(int i, string y)</code>	Inserts string y at index i	<code>a.Insert(1, "x");</code>	<code>"cxool"</code>
<code>Remove(int i, int count)</code>	Removes count number of characters from the string, starting from and including the character at index i	<code>a.Remove(1, 1);</code>	<code>"col"</code>
<code>Replace(string target, string substitute)</code>	Replaces all occurrence of target with substitute	<code>a.Replace("o", "x");</code>	<code>"cxxl"</code>
<code>Trim()</code>	Removes all leading and trailing whitespaces	<code>" yey ".Trim();</code>	<code>"yey"</code>
<code>ToUpper()</code>	Turns every character into upper case	<code>a.ToUpper();</code>	<code>"COOL"</code>
<code>ToLower()</code>	Turns every character into lower case	<code>"HaX".ToLower();</code>	<code>"hax"</code>
<code>ToCharArray()</code>	Converts the string into an array of characters	<code>a.ToCharArray();</code>	<code>{'c', 'o', 'o', 'l'}</code>
<code>String(char[])</code>	Converts an array of characters into a string	<code>string c = new String( {'x', 'y'} );</code>	<code>"xy"</code>

# File processing

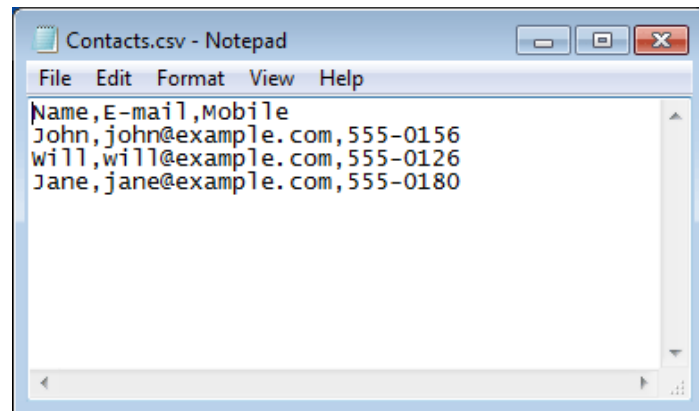
- Write a program to create multiplication table 10x10 and write it to “multi.txt” in the current folder
- Make 10 a parameter in your program to be able to change it
- Extend to 1000x1000 table and see the file size
- Can you try 100kx100k?
- Now change your program to flush after every 10k results



# CSV file



- Comma Separated File or CSV file is a simple format used to store tabular data
  - Similar to spreadsheet (Microsoft Excel, OpenOffice calc,...)
  - Store database
- Lists of items
- Each line is a row in table or item
- The number of items in rows could be different (Jagged Array)
- Each column has different values of parameters for that item
- It could be used for any general tabular format



```

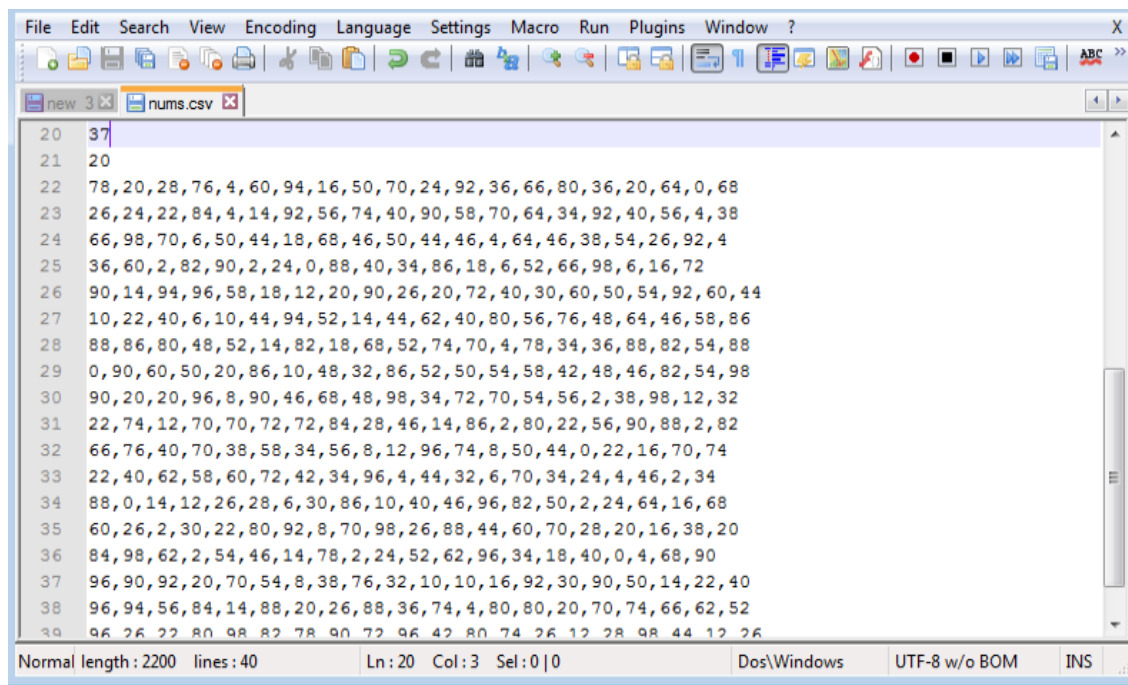
7 4107, 4316, 4219, 4105, 4105, 3918, 3873, 3819, 3817, 3871, 3931, 3874, 3851, 3849, 3881, 3909, 3999, 4146, 4329, 4394, 4066
8 4256, 4579, 4434, 4160, 3954, 3828, 3809, 3788, 3769, 3764, 3807, 3810, 3820, 3826, 3823, 3827, 3966, 4214, 4500, 4651, 4237
9 4351, 4642, 4537, 4223, 3937, 3859, 3880, 3836, 3791, 3750, 3741, 3807, 3821, 3859, 3882, 3822, 3936, 4244, 4556, 4643, 4375
10 4300, 4442, 4549, 4219, 3903, 3891, 3906, 3878, 3793, 3771, 3750, 3786, 3809, 3867, 3880, 3871, 3901, 4223, 4537, 4431, 4284
11 4321, 4555, 4480, 4185, 3902, 3824, 3868, 3853, 3785, 3762, 3711, 3733, 3777, 3802, 3826, 3830, 3894, 4171, 4443, 4544, 4270
12 4171, 4557, 4388, 4076, 3867, 3797, 3796, 3816, 3801, 3793, 3792, 3759, 3755, 3763, 3755, 3768, 3844, 4019, 4300, 4473, 4181
13 4082, 4316, 4194, 3941, 3817, 3795, 3828, 3817, 3827, 3873, 3909, 3854, 3782, 3766, 3803, 3790, 3835, 3910, 4093, 4218, 4128
14 4178, 4177, 4033, 3874, 3886, 3949, 3945, 3941, 3898, 3940, 4010, 3939, 3850, 3875, 3888, 3990, 4067, 3971, 4032, 4146, 4275
15 4149, 4309, 4097, 4008, 4133, 4268, 4197, 4036, 4010, 4086, 4155, 4081, 3974, 3976, 4097, 4338, 4378, 4230, 4220, 4418, 4221
16 4108, 4692, 4407, 4295, 4428, 4451, 4228, 4041, 4161, 4377, 4469, 4402, 4097, 4014, 4174, 4483, 4529, 4444, 4492, 4750, 4175
17 4100, 4102, 4148, 4266, 4315, 4160, 4005, 3753, 3900, 4221, 4414, 4137, 3892, 3756, 4023, 4168, 4286, 4329, 4221, 4158, 4138

```



# Reading a sample CSV file

- Suppose that we want to read a table  $m \times n$  stored in a CSV file like bellow and find max
- The first two lines has number of rows and column of a regular table start from line 3
- It is not the standard case that we keep number of rows and columns



```
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 3 x nums.csv x
20 37
21 20
22 78,20,28,76,4,60,94,16,50,70,24,92,36,66,80,36,20,64,0,68
23 26,24,22,84,4,14,92,56,74,40,90,58,70,64,34,92,40,56,4,38
24 66,98,70,6,50,44,18,68,46,50,44,46,4,64,46,38,54,26,92,4
25 36,60,2,82,90,2,24,0,88,40,34,86,18,6,52,66,98,6,16,72
26 90,14,94,96,58,18,12,20,90,26,20,72,40,30,60,50,54,92,60,44
27 10,22,40,6,10,44,94,52,14,44,62,40,80,56,76,48,64,46,58,86
28 88,86,80,48,52,14,82,18,68,52,74,70,4,78,34,36,88,82,54,88
29 0,90,60,50,20,86,10,48,32,86,52,50,54,58,42,48,46,82,54,98
30 90,20,20,96,8,90,46,68,48,98,34,72,70,54,56,2,38,98,12,32
31 22,74,12,70,70,72,72,84,28,46,14,86,2,80,22,56,90,88,2,82
32 66,76,40,70,38,58,34,56,8,12,96,74,8,50,44,0,22,16,70,74
33 22,40,62,58,60,72,42,34,96,4,44,32,6,70,34,24,4,46,2,34
34 88,0,14,12,26,28,6,30,86,10,40,46,96,82,50,2,24,64,16,68
35 60,26,2,30,22,80,92,8,70,98,26,88,44,60,70,28,20,16,38,20
36 84,98,62,2,54,46,14,78,2,24,52,62,96,34,18,40,0,4,68,90
37 96,90,92,20,70,54,8,38,76,32,10,10,16,92,30,90,50,14,22,40
38 96,94,56,84,14,88,20,26,88,36,74,4,80,80,20,70,74,66,62,52
39 96,26,22,80,98,82,78,90,72,96,42,80,74,26,12,28,98,44,12,26
Normal length: 2200 lines: 40 Ln: 20 Col: 3 Sel: 0|0 Dos\Windows UTF-8 w/o BOM INS
```



# Reading Files – CSV File Processing

```
int[ ][ ] data; //
// Read the array of intensities from the CSV file
FileStream inCSVFile = new FileStream(@"input.csv", FileMode.Open, FileAccess.Read);
StreamReader sr = new StreamReader( inCSVFile );

int rows = int.Parse( sr.ReadLine( ) ); // read first line, convert to int rows
int cols = int.Parse( sr.ReadLine( ) ); // read second line, convert to int cols

data= new int[ rows ][ ]; // allocate correct # of rows to main array
for( int row = 0; row < rows; row ++ )
{
    data[ row ] = new int[ cols ]; // allocate correct # of cols to each row
    string[ ] words = sr.ReadLine( ).Split( ",".ToCharArray( ) );
    for( int col = 0; col < cols; col ++ )
    {
        data[ row ][ col ] = int.Parse( words[ col ] ); // assign each value into array cell
    }
}
// calculate the max
```

# Practice: Simpsons and Programming

- Download zip file “SimpsonsProgramming.rar” from one drive and extract in your project folder
- It has some submitted C# program from The Simpsons family
- In the header of each submitted file there is information about the person submitted the file
- Open files one by one, read info, and write the following items in “out.csv”:
  - First name, last name, #std, number of lines in the submitted cs file
- The winner is the person with more codes
- Improve your program:
  - Ignore the blank lines in program and count the lines with at least one character (even “{“)
  - Find the number of lines that are not comment



# Practice Problem: Shakespeare's Writing

- Download CompleteWorksOfShakespeare.txt from OneDrive folder
- Read the file and
  1. count the number of lines & total number of words
  2. count the number of occurrence of **thy** and **'tis** in the words
  3. count the number of occurrence of words beginning with **well-**
    - eg well-deserved, well-derived
  4. count the number of occurrence of words ending with **ly**
  5. count the number of occurrence of words that has a single **-** in the middle (not first and not last character)
    - Make sure you aren't counting words with more than a single **-**



- All of his works are now available free online!
- <http://www.gutenberg.org/ebooks/100>

## WA5:

## ASSIGNMENT

The file “14522-8.txt” contains a Project Gutenberg eBook called “The Canterville Ghost”, by Oscar Wilde. Do not alter the file to complete this assignment.

Write a C# program which will read the file and show how often the author uses words of each length. A word is counted each time it is used, whether or not it has already been seen. You may assume in advance that there are no words longer than 25 characters.

In processing the file, consider only the lines after but including the line containing “THE CANTERVILLE GHOST” and before but including the line containing “Virginia blushed.”

In splitting each line into words, split around spaces (normal word separator) and hyphens (to break up compound words).

Before determining the length of a word, trim from each end, any extra space and any of the following characters: comma, period, question mark, exclamation mark, semicolon, colon, quote, double quote, left bracket, right bracket, left parenthesis, right parenthesis, vertical bar, underscore, plus sign, hyphen.

Any console output is acceptable which shows each word length and the number of times a words of that length appears. At a test case, I find 2848 uses of a word of length three.

In testing your file reading and division into words, you may see a couple of words which don't display properly in the console window because they use a Unicode character outside the normal range. This is just a display problem which you can ignore. An example is the word “mediaevalism” where the fifth character may not display properly. It will still have length eleven.

A couple of Oscar Wilde quotes related to learning:

*“Education is an admirable thing, but it is well to remember from time to time that nothing that is worth knowing can be taught.”*

*“We teach people how to remember, we never teach them how to grow.”*

*“Experience is the name everyone gives to their mistakes.”*

*“The truth is rarely pure and never simple.”*

# THE END ... ???

---

No, We have a talk about cryptography and cryptanalysis

# Practice Problem: Shakespeare's Writing

- Download CompleteWorksOfShakespeare.txt from OneDrive folder
- Read the file and
  1. count the number of lines & total number of words
  2. count the number of occurrence of **thy** and **'tis** in the words
  3. count the number of occurrence of words beginning with **well-**
    - eg well-deserved, well-derived
  4. count the number of occurrence of words ending with **ly**
  5. count the number of occurrence of words that has a single **-** in the middle (not first and not last character)
    - Make sure you aren't counting words with more than a single **-**



```
cmd.exe
C:\Users\Jeff\Desktop\BME121>Shakespeare
The Complete Works of Shakespeare has 124,787 lines of text and 1,410,671 words.

Number of thy : 4028
Number of 'tis : 1367
Number of words starting with well- : 91
Number of words ending with ly : 4211
Number of words with a single - in the middle : 4574

C:\Users\Jeff\Desktop\BME121>
```

- All of his works are now available free online!
- <http://www.gutenberg.org/ebooks/100>