# PowerShell II.

**Summary:** In this lesson we will learn what are rules of the base program blocks usage. How you can organize different kind of branches and cycles. You will see how you can use the main mathematical functions and other possibilities offered by .NET. We shall learn how to create a custom function and how to parameterize it. We shall see how you can read data from the keyboard and how you can write a special filter script.

**Commands**: *read-host (read), Test-Path*
**Structures**: *if (if), switch (case), for (for- C++ like), foreach (similar to for in shell), while (while), do-while, function name(listed params){instructions}*
**.Net functions:** *[Math]::function() (e.g. [Math]::sqrt(value)), string.split() (splitting up a string into pieces), string.substring() (similar to cut)*
**Variable:** *$input (special variable for standard input)*

## Base program structures

a) Implement a PowerShell script which counts the factorial value a given number N (N!)! The number is given by a parameter! Check whether there is a parameter or not! (if, for, $args)
b) Modify it and use param([int]$N) for parameter handling!
c) Create another version and use a mandatory parameter syntax! (Param( [Parameter(Mandatory=$true)] [int]$N)
d) Create a PowerShell script which decide whether a number is a prime number or not!
e) Implement a PowerShell script which evaluate the sum of the parameters (we do not know how many parameters will be given)! (for, $args.length or foreach)
f) Create a PowerShell script which decide whether a number is a prime number or not! The number is going to be read from the keyboard! (while, %)

## Functions

g) Write a PowerShell script which gives back the results of a quadratic equtation ax2+bx+c=0! The coefficients are given by parameters! You can use [Math]::sqrt() function to extracting the root!
h) Modify the PowerShell script and read the coefficients from keyboard! Be careful with the required types! Maybe you have to use typecasting too. (read-host, if, write-host)
i) Create a PowerShell script which decide whether a number is a prime number or not! Write a function and call it!
j) Implement a PowerShell script which gives back all the prime numbers inside an interval. The ends of the interval are read from the keyboard! (read-host, while, %, for)

## Files

k) Write a script which gets a file and writes out into two files the even-th and odd-th lines! The name of the original file will be given by a parameter, the name of the two result files should be: even.txt and odd.txt! Check whether the file is existing or not! (get-content, for, test-path - - In Unix while ..< file, test -f)
l) Create a PowerShell script which gets a lot of coded e-mail addresses in a file and tranfer them to the standard format! E.g. somebody at inf dot elte dot hu into

*Financed from the financial support ELTE won from the Higher Education Restructuring Fund of the Hungarian Government*

somebody@inf.elte.hu. The coded file is given by parameter. If the script does not get a parameter, read the file-name from the keyboard! Check whether it is existing or not! The result must be copied into the result.txt file and listed on the screen too! (string.replace() - In Unix sed or awk)

m) Write a script which writes out the content of a file in reverse order! The first line at last…(use a string array as the content of the file)
n) Create a script which writes out in reverse order the lines of the file! (The first character at the last position of the line..) (use line as string and for - In Unix rev or cut…)
o) Write a script which changes the order of the words in each line. You get the filename from the parameterlist and the lines in the file contain 3 words. (split() - In Unix awk )
p) Modify the above created script to be able to work as a filter command! (Special variable name: $input – it is the same as `cat` in UNIX.)