

מחשוב מקבילי ומבוזר

תרגיל # 2

The purpose of this exercise is to practice MPI Cartesian topology

Problem definition

Launch K^2 processes. The process 0 reads K^2 strings from the given file **data.txt**. Each string has the same length $2*N$. The process 0 distributes these strings between all process in such a way that process with rank i receives the i -th string.

Perform following iteration steps:

1. Organize the processes using $K \times K$ Cartesian two-dimensional topology.
2. Each process sends the characters from odd places of its current string to the left process and the even located characters to the up process (allow each process to have all four neighbors).
3. The characters received from the right process became a first part of the current string, The characters received from the down process became a second part of the current string (Fig.1, Fig.2).
4. Each process searches if the given string **SubString** presents into the current string. If the answer is positive for at least one process - the program ends and the process 0 displays all string received from each process according to their ranks – in order from the first to the last.
5. If the **SubString** was not found by any process – return to the step 2 until the maximum iteration **MaxIterations** is performed.
6. If the **SubString** was not found after **MaxIterations** - the process 0 outputs “The string was not found”.

For example, the current string **a b c d** will be substituted by the string **l n x z** at the next iteration.

	a b c d	k l m n
	x y z w	

Fig.1 Iteration I

	l n x z	

Fig.2 Iteration I + 1

Input File structure:

K N MaxIterations
SubString
String1
String2
String3
...
StringK²

Grading Policy:

- **10 points** for code quality:
 - a. The code must be divided into small functions (not more than 40 lines of code).
 - b. Use meaningful names for variables, functions, files, constants.
 - c. Place enough comments to understand the code
 - d. No unused lines of code. Don't repeat the code – use functions!
 - e. Write README.TXT file if special instructions are needed to run the solution. The file must be in the root folder of the solution.
- **90 points** – for proper implementation of the requirements.

Important:

- The Homework has to be tested under Ubuntu OS in VLAB with compilation and run from the **Eclipse**.
- Supply the whole compressed directory of your Eclipse project.
- The Homework must be delivered in time. No delay will be accepted. It may be performed in pairs. Only one member of pair submits the solution through the Moodle.
- The whole solution must be zipped and named as

11111111_22222222.zip

Where **11111111** is ID of the one student and **22222222** is ID of another student