**מחשוב מקבילי ומבוזר**

תרגיל #5

**The purpose of this exercise is to practice the MPI Cartesian Topology,**

**Scatter, Gather and Sort**

Write a parallel program to implement **Shearsort Algorithm** to sort a set of cuboids (תיבה) bodies in ascending order.

# Requirements:

* One of the processes reads all data from the text file **cuboids.dat**. Each line represents one cuboid – its id, length, width, height. The number of lines (bodies) is equal to the number of processes launched.
* This process will display result of the sort and writes it to the file **result.dat**. It uses **Scatter** to send data to processes, **Gather** to collect result of sorting.

* Use **Cartesian** Topology for communication between processors during sorting.
* Implement **Shear Sort** to sort the bodies. The bodies are sorted according to their volumes. In case of equal volumes, the bodies are compared according to their heights.
* Use **Odd Even Sort** to sort rows and columns. Assume that number od rows and columns is even. In other word, the number of processes (bodies) can be represented as n = (2k)2
* The input file **cuboids.dat** is organized as following:

0 6 6 6

1 7 8 7

2 2.3 2 2

3 2 3 2

4 4 2 3

5 1.9 4 3

6 9 9 9

7 7 7 8

8 5 5 5

9 1.2 1.3 1.56

10 7 7 7

11 1.01 2.9 1

12 3 2 4

13 8 8 8

14 2.1 3.1 2.2

15 9 8 9

* The output file **result.dat** contains ids of bodies according to sorted criteria

For example

9 11 2 3 14 5 4 12 8 0 10 1 7 13 15 6

**Note:**

You need (2log(n)+1) row/column phases to sort n2 cuboids.

# Grading Policy:

* **10 points** for code quality:
  1. The code has to be divided into small functions (not more than 40 lines of code).
  2. Use meaningful names for variables, functions, files, constants.
  3. Place enough comments to understand the code
  4. No unused lines of code. Don't repeat the code – use functions!
  5. Write **README.TXT** file if special instructions are needed to run the solution. The file must be in the root folder of the project.
* **90 points** – for proper implementation of the requirements.
* The Homework must be delivered in time. No delay will be accepted.

# Important:

* The homework may be performed in pairs. Only one member of pair submits the solution through the Moodle. The whole project must be zipped and named as

**111111111\_222222222.zip**

Where **111111111** is ID of the one student and **222222222** is ID of another student

בהצלחה!