PWr Week 6

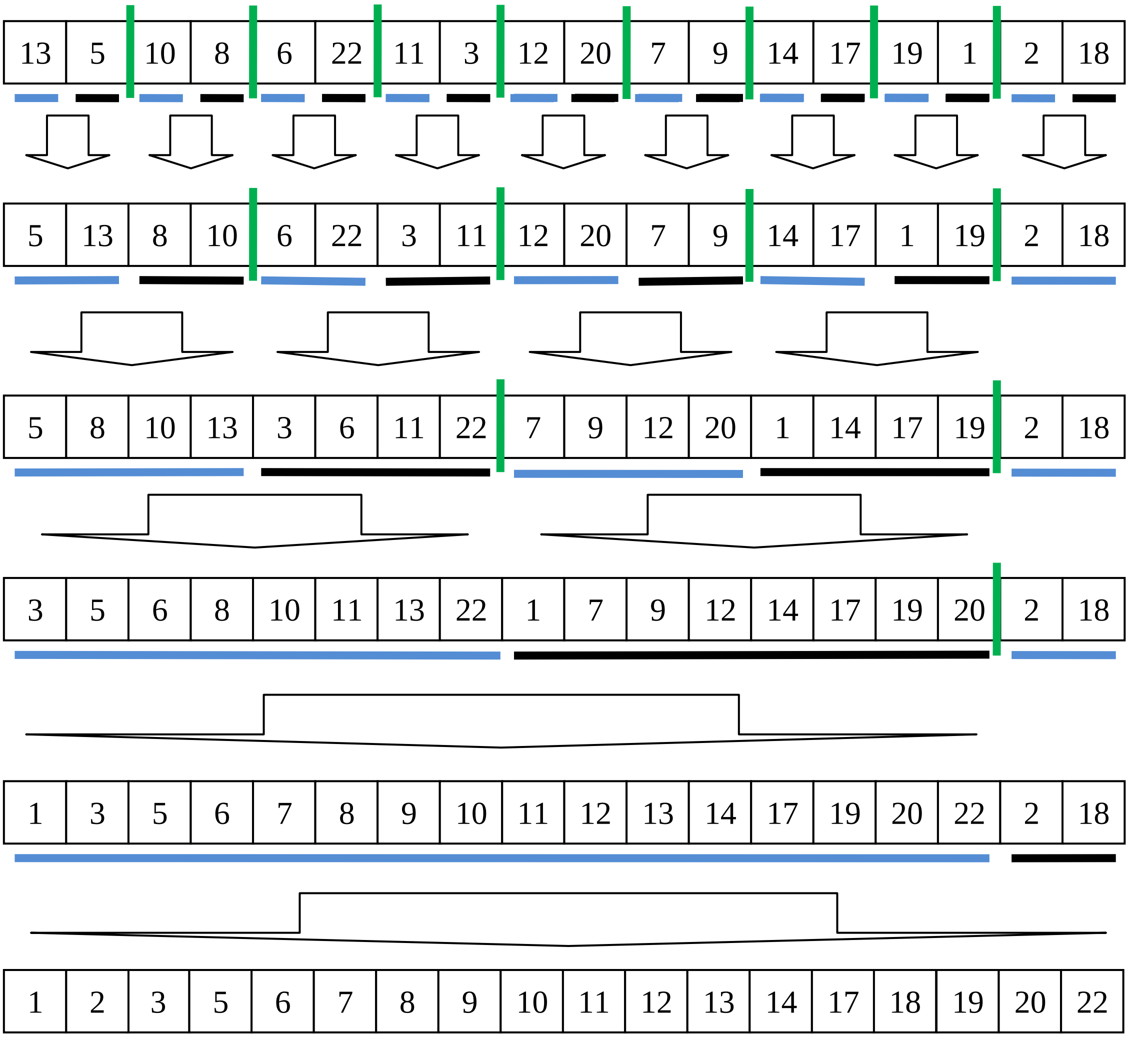
Algorithms and Data Structures

Laboratory – **List 6**

1. Implement iterative version on mergesort algorithm. Choose a version in which first we assume every one-element array is sorted. Then we merge every pair of consecutive one-element arrays into two-element sorted arrays. In next step every pair of consecutive two-element arrays into four-element sorted arrays, and so on. which present the array in one line: after every value put a comma. Implement function `void mergeSortIter(int array[], int size)` for sorting in increasing order. Print state of the array after each execution of outer loop.

For **10 points** present solutions for this list till **Week 7**.
For 8 **points** present solutions for this list till **Week 8**.
For 5 **points** present solutions for this list till **Week 9**.
After Week 9 the list is closed.

Graphical explanation for iterative mergesort:



13 5 | 10 8 | 6 22 | 11 3 | 12 20 | 7 9 | 14 17 | 19 1 | 2 18

5 13 8 10 | 6 22 3 11 | 12 20 7 9 | 14 17 1 19 | 2 18

5 8 10 13 3 6 11 22 | 7 9 12 20 1 14 17 19 | 2 18

3 5 6 8 10 11 13 22 1 7 9 12 14 17 19 20 | 2 18

1 3 5 6 7 8 9 10 11 12 13 14 17 19 20 22 2 18

1 2 3 5 6 7 8 9 10 11 12 13 14 17 18 19 20 22

Appendix 1

The solution of this task will be automated tested with tests from console of presented below format. This is an extension of main program from previous list. You need only to add one command:

If a line has a format:

`MI` *n*

in next lines there will be *n* integer numbers separated by space or newline. The array have to be sorted using `mergeSortIter` function, writing on the console state of array after every big step of the algorithm. Before start the function write in one line “mergeSortIter” and initial state of the array in next line.

For example for input test:

```
MI 18
13 5 10 8 6 22 11 3 12 20 7 9 14 17 19 1 2 18
HA
```

The output have to be:

```
mergeSortIter
13,5,10,8,6,22,11,3,12,20,7,9,14,17,19,1,2,18,
5,13,8,10,6,22,3,11,12,20,7,9,14,17,1,19,2,18,
5,13,8,10,6,22,3,11,12,20,7,9,14,17,1,19,2,18,
5,8,10,13,3,6,11,22,7,9,12,20,1,14,17,19,2,18,
3,5,6,8,10,11,13,22,1,7,9,12,14,17,19,20,2,18,
1,3,5,6,7,8,9,10,11,12,13,14,17,19,20,22,2,18,
1,2,3,5,6,7,8,9,10,11,12,13,14,17,18,19,20,22,
END OF EXECUTION
```