

Project 2**Handed out:** 30.10.2018**Due:** 26.11.2018, until 23.30

1. PROBLEM

In this project, you are going to analyze the performance of a call center's employees. Using the heapsort algorithm, you will report the employees with the best/worst performance score and with the maximum/minimum number of calls.

2. DATASET

You are provided a dataset consists of 11 csv files. The first 10 files (day1 to day10) contain data for 10 different days. The employee ID, total number of calls received, number of calls with positive feedback and negative feedback are given in the columns of these files, respectively. The records in files are sorted in ascending order according to IDs. The following formula is used to calculate the performance scores of the employees:

$$PS = 2 * (\text{number of calls}) + (\text{number of positive feedback}) - (\text{number of negative feedback})$$

In the dataset, there is another file, "numbers.csv". This file is used for some additional analyses explained in Section 4.

3. IMPLEMENTATION (60 POINTS)

You will implement the following procedures detailed in the lecture notes:

- `max/min-heapify()`
- `build-max/min-heap()`
- `heapsort()`
- `insert()`
- `extract-max/min()`
- `increase-key()`

In your program, first, the necessary heaps will be constructed by using the data in "day1.csv" file. Then, the heaps will be updated by reading the other csv files in order. These updates will be performed right after each line in the files is read. If the employee ID read from the file is present in the heap, the employee's performance score and the number of calls will be increased using the `increase-key()` procedure. If the employee ID is not present in the heap, it will be added to the heaps using the `insert()` procedure. Since the employee ID is in ascending order, if an ID is larger than the ones previously encountered, this ID should be considered as a new ID (not in the heaps). After the necessary operations for each csv file (for each day) are completed, your program is expected to report the employee IDs as follows:

```
AFTER DAY-1
BEST PERFORMANCE : 123, 456, 789
WORST PERFORMANCE: 321, 654, 987
MAXIMUM CALLS     : 111, 222, 333
MINIMUM CALLS     : 444, 555, 666
```

```
AFTER DAY-2
BEST PERFORMANCE : 246, 135, 159
```

Project 2

```
WORST PERFORMANCE: 642, 531, 951
MAXIMUM CALLS      : 111, 333, 999
MINIMUM CALLS      : 555, 444, 888
```

```
AFTER DAY-3
```

```
...
...
```

4. REPORT (40 POINTS)

- a. (10 points)** Give a detailed explanation of your implementation for each step.
- b. (15 points)** Give the running time for each procedure mentioned above and show that the procedures you implemented are in accordance with these values.
- c. (15 points)** Sort the numbers in the “numbers.csv” file in 10 steps by selecting the largest 200000 numbers in each step, and plot the following values:
- The height of the heap before each step
 - The execution time of each step.

Save the sorted numbers into a file to make sure the program is working correctly.

5. DETAILED INSTRUCTIONS

- Your program must read all the csv files from the same directory with your source code.
- All your code must be written in C++ using object oriented approach and able to compile and run on Linux using g++.
- Do not use external libraries such as STL.
- Submissions will be done through the Ninova server. You must submit all your program and header files. You must also submit a softcopy report.
- Each student must work individually for the project. Team work is not accepted!

If you have any questions, please feel free to contact Res. Asst. Emrah Başaran via e-mail (basaranemrah@itu.edu.tr).