

ASSIGNMENT 1

Subject: Recursions

Due Date: 02.11.2021

1 Introduction

Alphametic: An alphametic puzzle (also known as a *cryptarithm*) is a type of puzzle where words are put together into an arithmetic formula such that digits can be substituted for the letters to make the formula true. The main goal here is to assign each letter to a digit from 0 to 9. Each digit must be assigned only one letter, that's to say, not any digit can be assigned different letters.

$$\begin{array}{rcccc} & B & A & S & E \\ + & B & A & L & L \\ \hline G & A & M & E & S \end{array}$$

If the puzzle above is considered, some letters were used multiple times and totally there are seven different letters must be found. To solve a puzzle there are one rule should be considered; There must be a one-to-one mapping between letters and digits. In other words, if you choose the digit 7 for the letter B, then all of the B's in the puzzle must be 7 and no other letter can be 7.

2 Experiment

Puzzle Solver: In this assignment you will create the algorithm that uses recursions to assign 0-9 digits to the letters. The algorithm consists of two processes: *puzzle creation* and *puzzle solving*. In the *puzzle creation* process, user enters the words as illustrated in Fig. 1 and the algorithm should consider the following conditions:

- The length of all words must be greater than or equal to 2
- The length of all words must be less than or equal to 6.
- Total number of different letters in all words must be less than or equal to 10.

Once the conditions above are met, the algorithm can proceed to the next process (*puzzle solving*).

In this process, the algorithm should try every possible digit for a letter by brute force approach to solve the puzzle. Thus, it will have tried every occurrence for whole letters at the end of the search process. (Note: At first glance using iterative method may seems reasonable; even so, you are obligated to use another way in the assignment, namely recursive method).

```
Enter 1st input value: DONALD
Enter 2nd input value: GERALD
Enter Output value:   ROBERT
```

Figure 1: Screenshot of initial window

Depending on the puzzle structure user created, the algorithm starts to try every possible solution for letters then displays them. Structurally, *puzzle solving* process consists of two parts, one of them produces the candidate solutions and the other checks if it works out correctly. To validate whether the algorithm works correctly or not, you are given below a benchmark test and its result found by the algorithm.

$$\begin{array}{r} D O N A L D \\ + G E R A L D \\ \hline R O B E R T \end{array}$$

Once you have completed your algorithm, you should expect to see the results given below.

```
searching...found!
DONALD: 526485
GERALD: 197485
ROBERT: 723970
```

As seen above, a puzzle as here has just one solution, in some other cases one puzzle may have multiple solutions. The algorithm has also to find the other solutions (if any).

Notes

- Do not miss the deadline.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate or very similar assignments are both going to be considered as **cheating**.
- You can ask your questions via Microsoft Teams and you are supposed to be aware of everything discussed in Microsoft Teams.
- You will submit your work as below:
→ <student_number>.java