Programming C++

Project **Kaprekar Constant**

Status: 13. Mai 2022, 17:36

Supervisor: Prof.Dr. G. Haase, gundolf.haase@uni-graz.at

The Indian mathematician D.R. Kaprekar[1] found a constant named after him which can be achieved using the rule described in Wikipedia[2]. There are unique constants for 3- and for 4-digit numbers with base 10, but we get non-unique constants and/or non-unique cycles for all other numbers of digits with base 10.

The final goal of this project consists in determining the constants and cycles for 2-20 digit numbers in base 10 and afterwards in an arbitrary base.

1. Implement the rule for 3 or 4 digit numbers of base 10. One possible course of action:

   - Write function(s) that generate the smallest and the largest number from a given number n with digits *abcd* contained in the given base.
   - Compare your constants with those from literature.
   - Take care that the result of, e.g. $221 - 122$ is again interpreted as 3-digit number. Similar with more digits.

   Storing the numbers:

   - You might store your numbers as `int` or `unsigned long long int` and extract the digits from this representation.
   - On the other hand you might store the numbers in its digits representation and provide the neccessary funtionality (min, max, substract, ...). A (template?) class would encapsulate that.

2. Extend you algorithm such that cycles as well as constants are determined for others than 3 or 4 digits. (6 Pkt.)

   - Use 2-10 digits[3] in base 10.
   - Use 11-20 digits[4] in base 10.
     Data type `long long int` fully includes up to 17 digits.
   - More than 20 digits?

3. Extend your algorithms to an arbitrary base. Check the results for special bases with available results[5]. (+2 Pkt.)

[1] `https://en.wikipedia.org/wiki/D._R._Kaprekar`

[2] `https://de.wikipedia.org/wiki/Kaprekar-Konstante#Verfahren_zur_Berechnung_der_Kaprekar-Konstante`

[3] `http://kaprekar.sourceforge.net/output/sample.php`

[4] `http://kaprekar.sourceforge.net/output/sample2.php`

[5] `https://en.wikipedia.org/wiki/Kaprekar's_routine#Kaprekar's_constants_and_cycles_of_the_Kaprekar_mapping_for_specific_base_b`