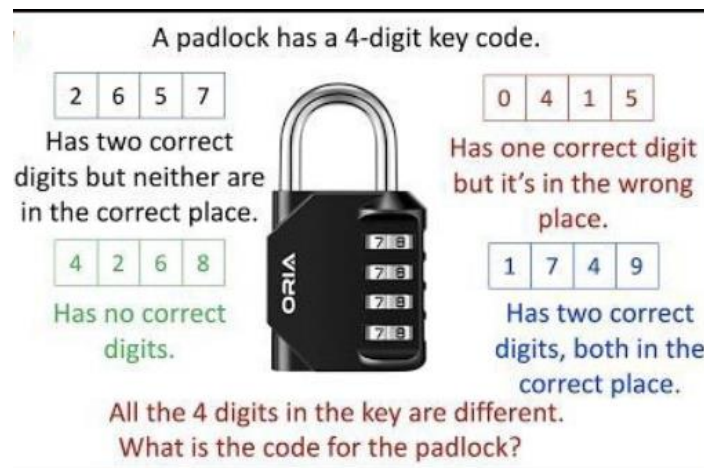


Challenge Nov-2022

Quick Solution using IBM ODM by Andrew Macdonald

macdonald@uk.ibm.com



This follows my usual design principles to keep things simple for the business rule authors and to match the given requirements and data structures where possible. I implemented this as a self-contained ruleset that has no input/output parameters. It was built to solve the problem rather than provide a flexible and extensible decision service.

For this challenge it just needs one rule and one helper function to count the number of exact and partial matching numbers :

definitions

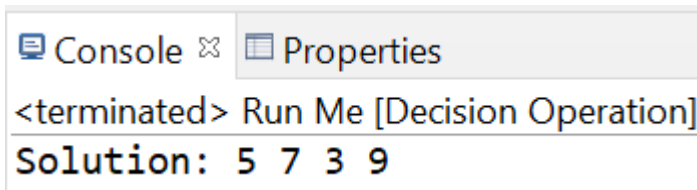
```
set Digits to { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 } ;
set Digit1 to a number in Digits ;
set Digit2 to a number in Digits
    where this number is not Digit1 ;
set Digit3 to a number in Digits
    where this number is not one of { Digit1 , Digit2 } ;
set Digit4 to a number in Digits
    where this number is not one of { Digit1 , Digit2 , Digit3 } ;

if
    the match score of Digit1 Digit2 Digit3 Digit4 to "2657" is 2.0
    and the match score of Digit1 Digit2 Digit3 Digit4 to "0415" is 1.0
    and the match score of Digit1 Digit2 Digit3 Digit4 to "4286" is 0.0
    and the match score of Digit1 Digit2 Digit3 Digit4 to "1749" is 0.2
then
    print "Solution: " + Digit1 + " " + Digit2 + " " + Digit3 + " " + Digit4 ;
```

The first line of definitions section defines the list of possible numbers to use. The next four lines define the four unique digit variables which will generate a list of all permutations of the 4 digit codes.

The four 'if' conditions test the provided codes. I decided to add a utility method to count how many digits were an exact match and how many matched but in the wrong place returned in the form of a decimal number <wrong place total>.<correct place total>. This enabled me to express the rule in a very similar format to how the problem was described and to make it easy to use other sets of values.

You can verbalise the method in different ways to make it more readable or natural language like but this is fine for a quick test. The 'then' part of the rule only executes if all of the parts of the definitions section have successfully matched and all of the if conditions are true. It will execute for every successful match so it should find all valid solutions. Running the rule directly in the ODM Rule Designer gives the following result showing only one possible solution:



The helper method is created as follows:

General Information

Name: matchScore

Type: double

Class: Helpers

☒ Static

Member Verbalization

✖ Remove

the verbalization.

✚ Create

a navigation phrase.

Navigation : "the match score of a number a nun"

Template:

the match score of {0} {1} {2} {3} to {4}

Arguments

Edit the arguments of this member.

Name	Type
d1	int
d2	int
d3	int
d4	int
combination	java.lang.String

The code of this method is as follows. This is just a quick and simple first attempt but it works fine for the challenge. It could obviously be improved by using an array type and loops but it served it's purpose in making the rule simple to author by the non-technical users.

```
int wrongplace = 0;
if (!combination.substring(0,1).equals(""+d1) && combination.contains(""+d1) ) wrongplace = wrongplace + 1;
if (!combination.substring(1,2).equals(""+d2) && combination.contains(""+d2) ) wrongplace = wrongplace + 1;
if (!combination.substring(2,3).equals(""+d3) && combination.contains(""+d3) ) wrongplace = wrongplace + 1;
if (!combination.substring(3,4).equals(""+d4) && combination.contains(""+d4) ) wrongplace = wrongplace + 1;

int rightplace = 0;
if (combination.substring(0,1).equals(""+d1) ) rightplace = rightplace + 1;
if (combination.substring(1,2).equals(""+d2) ) rightplace = rightplace + 1;
if (combination.substring(2,3).equals(""+d3) ) rightplace = rightplace + 1;
if (combination.substring(3,4).equals(""+d4) ) rightplace = rightplace + 1;

return wrongplace+rightplace/10.0;
```

This report has been created by Andrew Macdonald and reflects his own views. It is provided for informational purposes only, and is neither intended to, nor shall have the effect of being, legal or other guidance or advice to any readers. While efforts were made to verify the completeness and accuracy of the information contained in this document, it is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this document or any other materials. Nothing contained in this document is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.