

Challenge Nov 2022

Crack Padlock Code

A solution with DT5GL by Jack Jansonius – 17 December 2022

Problem Statement (from the web site):

A padlock has a 4-digit key code.

2	6	5	7
---	---	---	---

Has two correct digits but neither are in the correct place.

4	2	6	8
---	---	---	---

Has no correct digits.



0	4	1	5
---	---	---	---

Has one correct digit but it's in the wrong place.

1	7	4	9
---	---	---	---

Has two correct digits, both in the correct place.

All the 4 digits in the key are different.
What is the code for the padlock?

Implementation of the decision model in DT5GL (Solution 1):

Remark:

In this case, the condition "All the four digits in the key are different" doesn't matter!

```
rTable 0:
If:                                     | 0| 1| 2|
Next cnumber in [0-9999]              | Y| Y| N|
True = checknr(cnumber, '2657', 2, 0) | Y| -| -|
True = checknr(cnumber, '4268', 0, 0) | Y| -| -|
True = checknr(cnumber, '0415', 1, 0) | Y| -| -|
True = checknr(cnumber, '1749', 2, 2) | Y| -| -|
Then:
Action is Printcode                   | X|   | 
Action is Nextcode                    |   | X|   |
Action is Finished                     |   |   | X|
# .....
# repeat until: Finished

Attribute: True Type: Integer
Equals: 1

GoalAttribute: Action
Repeat_until: Finished

Case: Finished
Print: "Finished!"

Case: Printcode
Print: "Code found: %s "    cnumber

Case: Nextcode
Print: "#REM# - "
```

Additionally, in another file.py:¹

```
def checknr(number, checknumber, nr_correct, nr_correct_position):
    count_nr_correct = 0
    count_nr_correct_position = 0
    lnumber = str(number).zfill(len(checknumber)) # make string with leading zeros
    for i in range(len(checknumber)):
        if lnumber[i] in checknumber:
            count_nr_correct += 1
        if lnumber[i] == checknumber[i]:
            count_nr_correct_position += 1
    return nr_correct == count_nr_correct and \
           nr_correct_position == count_nr_correct_position
```

Output program:

```
Code found: 5739
Finished!
```

¹ As also present in my solution for: <https://dmcommunity.org/challenge/challenge-sep-2019/>

Implementation of the decision model in DT5GL (Solution 2):

In the puzzle, if the fourth clue: '1749', 2, 2 (2 correct and in the right place) is changed to: '4759', 3, 2 (3 correct, 2 of which are in the right place), then the output of the program becomes:

```
Code found: 1779
Code found: 5739
Code found: 7709
Finished!
```

and that's interesting, because now the condition "All the four digits in the key are different" does matter!

Now the program becomes:

```
rTable 0:
If:                                     | 0 | 1 | 2 |
Next cnumber in [0-9999]               | Y | Y | N |
True = checknr(cnumber, '2657', 2, 0)   | Y | - | - |
True = checknr(cnumber, '4268', 0, 0)   | Y | - | - |
True = checknr(cnumber, '0415', 1, 0)   | Y | - | - |
True = checknr(cnumber, '4759', 3, 2)   | Y | - | - |
True = uniqueDigits(cnumber)           | Y | - | - |2
Then:
Action is Printcode                    | X |   |   |
Action is Nextcode                     |   | X |   |
Action is Finished                      |   |   | X |
# .....
# repeat until: Finished

Attribute: True Type: Integer
Equals: 1

GoalAttribute: Action
Repeat_until: Finished

Case: Finished
Print: "Finished!"

Case: Printcode
Print: "Code found: %s "    cnumber

Case: Nextcode
Print: "#REM# - "
```

In addition, in another file.py, in addition to the previously mentioned function checknr:

```
def uniqueDigits (cnumber):
    return len(str(cnumber)) == len(set(str(cnumber)))3
```

Output program:

```
Code found: 5739
Finished!
```

² Instead of this function call, the condition here could also be:

True = len(str(cnumber)) == len(set(str(cnumber)))

³ Ask ChatGPT: use Python to determine if an integer consists of unique digits.

<https://openai.com/>

Implementation of the decision model in DT5GL (Solution 3):

Finally, solution 3 which actually speaks for itself:

```
rTable 0:
If:
Next cnumber in [0-9999]
True = checknr(cnumber, '2657', 2, 0)
True = checknr(cnumber, '4268', 0, 0)
True = checknr(cnumber, '0415', 1, 0)
True = checknr(cnumber, '4759', 3, 2)
True = uniqueDigits(cnumber)
Then:
Action is Printcode
Action is Printcode_notUnique
Action is Nextcode
Action is Finished
# .....
# repeat until: Finished

Attribute: True Type: Integer
Equals: 1

GoalAttribute: Action
Repeat_until: Finished

Case: Finished
Print: "Finished!"

Case: Printcode
Print: "Code found: %s "    cnumber

Case: Printcode_notUnique
Print: "Code found: %s (not all different)"    cnumber

Case: Nextcode
Print: "#REM# - "
```

Resulting in:

```
Code found: 1779 (not all different)
Code found: 5739
Code found: 7709 (not all different)
Finished!
```