Exercise 2.1 (May 1, 2020)

-B9TB1707

Question:

Find all numbers of 3 digits such that the sum of the cubes of its digits equals the number itself; an example is 153, because $1^3+5^3+3^3 = 153$

Solution:

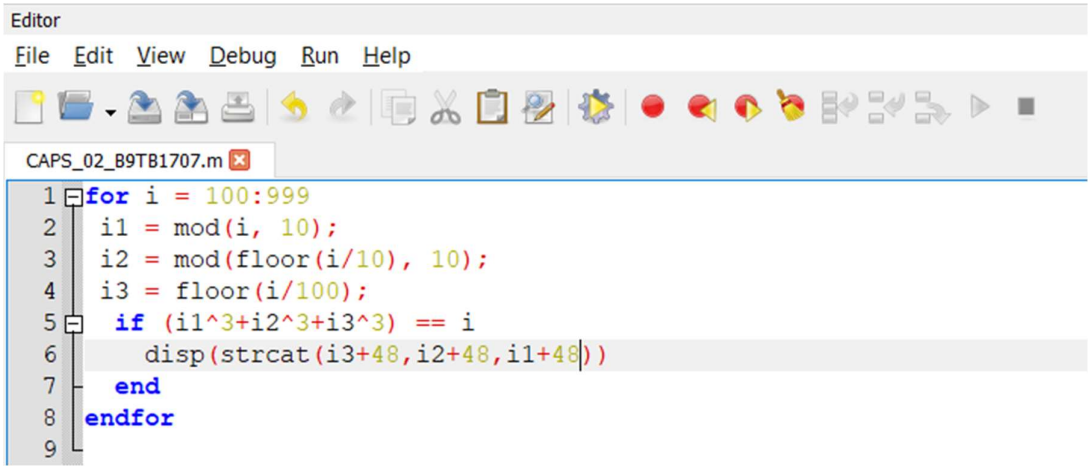
There are many ways to solve the question given in this week's class. We were tasked to write the code for this question in two ways based on the hints given.

Solution 1:

Given hint is as follows:

```
for i = 100:999
  i1 = mod(i, 10);
  i2 = mod(floor(i/10), 10);
  i3 = floor(i/100);
  disp([i3 i2 i1])
endfor
```

The preceding code relies on the looping over all three digit numbers and splits every integer into its constituent single digits.
To get the solution all I need to do is cube the individual digits and equality check with the original three digit integer.
Below is the finished code.



The output is as follows.

```
Command Window
>> CAPS_02_B9TB1707

153
370
371
407
>>
```

How it works:

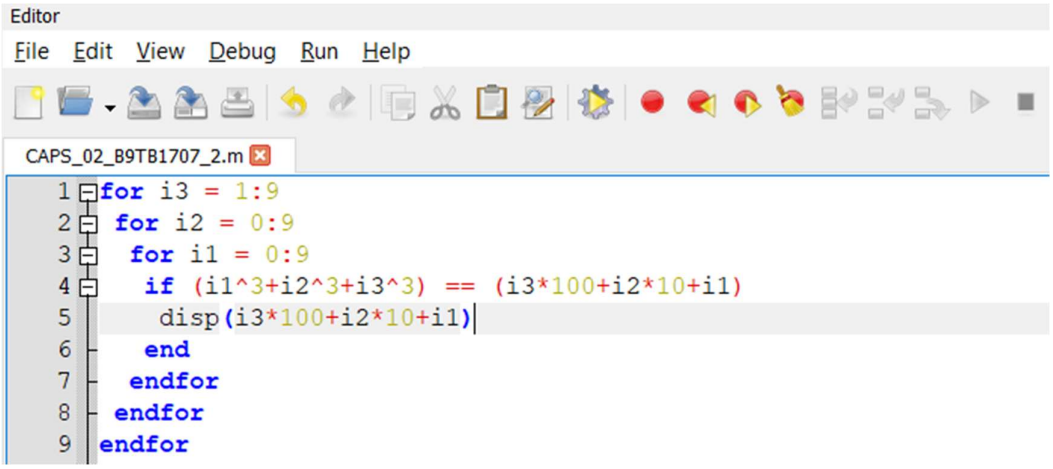
1. Line 1 sets up the for loop; iterating from 100 to 999 (every 3 digit number)
2. Line 2 gets the digit in the unit's place. Mod() function is function that returns the remainder, so modulus 10 returns the digit in the unit's place. The value is stored in variable i1.
3. Line 3 gets the digit in the ten's place. The process is similar to step 2, only difference is that we need to remove the unit's place. To do this I first shift the decimal place by dividing by 10 and then remove the decimal values by using the function floor(). Floor() returns the closest integer value smaller than or equal to the input. The value is stored in variable i2
4. Line 4 stores the digit in the hundred's place. We divide by 100 and use floor to get rid of the decimal places.
5. Line 5 uses an if statement to check the condition of the problem statement and displays the number if it returns true.
6. Line six is an output statement. It uses strcat() function to concatenate the ASCII values of the respective digits. +48 is used because 48 is the ASCII value for zero.
7. Line 7 & 8 end the if and for bodies respectively.

Solution 2:

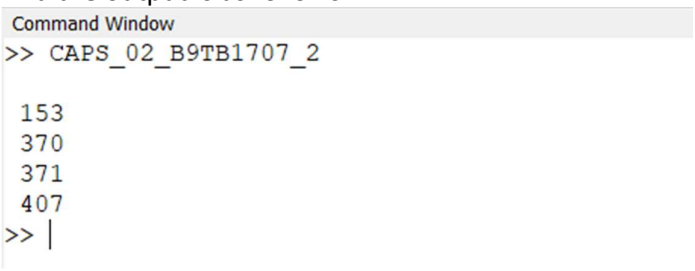
Given hint is as follows:

```
for i3 = 1:9
  for i2 = 0:9
    for i1 = 0:9
      [                    ]
    endfor
  endfor
endfor
```

The preceding code iterates through every 3-digit number by iterating through each digit in a nested for loop format. To get the solution all one needs to do is to perform an equality checking operation with an if statement. To retrieve the original number all one need is to multiply with place value with the appropriate face value.
The code below is my solution to the question.



And the output is as follows



How it works:

1. Line 1-3 establishes the for loops for the digits of the hundred's, ten's, and unit's place respectively.
2. Line 4 checks equality based on the condition given in the question. To get the original value all we need to do is multiply the appropriate face value and add.
3. Line 5 is the command for output if the if condition is true.
4. Line 6-8 are closing statements for the if and for bodies.

Conclusion:

Hence I have solved the given question in the both of the ways required. I feel that the second method is better. It both have the same time complexity, but the first method uses many function calls, which are eliminated in the second method. For this reason I believe a problem like this is better solved through the 2nd method.