

Joanna Masikowska

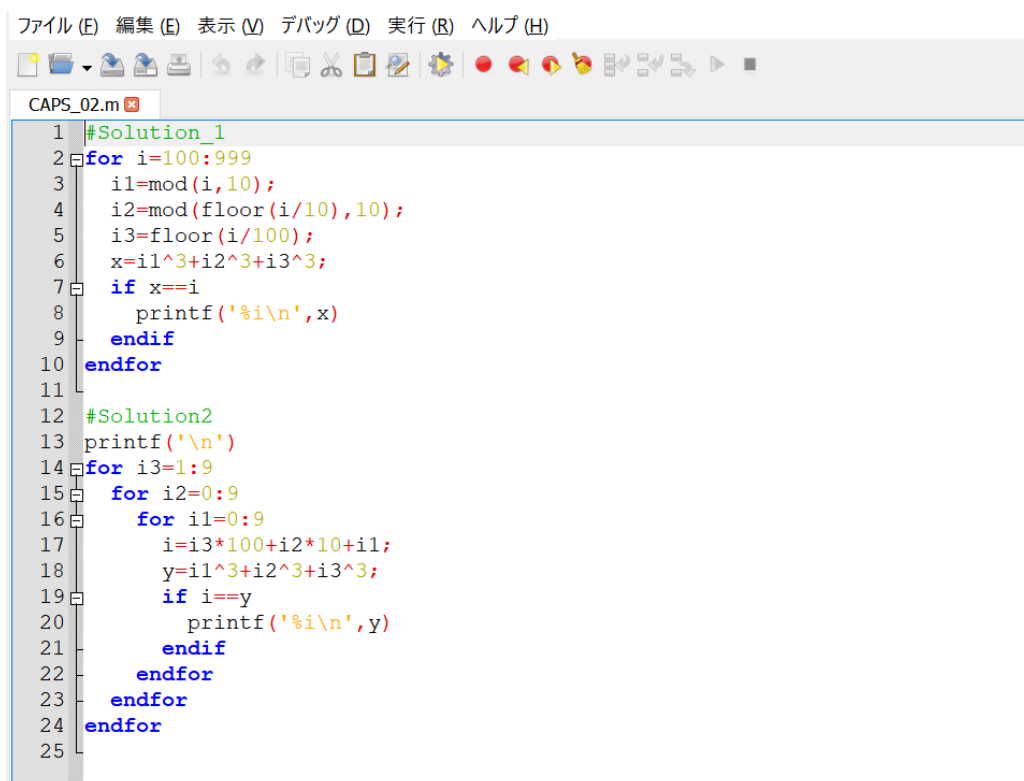
B9TB1710

#Solution1

I start my code with a **for** loop that goes through all 3-digit numbers (from 100 to 999), assigning a number to the variable **i** on each repetition. Inside the loop, variables **i1**, **i2**, **i3** that are assigned to the number of respectively: ones, tens, hundreds of **i**. To determine these variables at a given repetition of the loop, I use functions **floor** (which rounds the decimal number down to the integer) and **mod** (which returns the remainder). Then, I assign the sum of cubes of the digits of **i**, that is $i1^3+i2^3+i3^3$, to a variable **x**. Using **if** function, I compare variable **x** with **i**. If they are equal, function **printf** is activated and **x** is printed as an integer (because of **%i**). Otherwise, the program will not print anything. After the condition is checked, loop proceeds to the next repetition.

#Solution2

In this solution I use 3 loops. First loop, with variable **i3**, represents number of hundreds in a 3-digit number. Within it, there is a loop representing tens (with variable **i2**) and again within it there is a loop representing ones (variable **i1**). In the last loop I create variable **i** that is equal to a 3-digit number, and **y** that is equal to the sum of cubes of the digits the number. Then I use function **if** to check whether **i** and **y** have the same value. If they do, **y** will be printed as an integer. Otherwise, no additional command will be conducted. After the condition is checked, loop proceeds to the next repetition.



```
ファイル (F) 編集 (E) 表示 (V) デバッグ (D) 実行 (R) ヘルプ (H)
CAPS_02.m
1 #Solution_1
2 for i=100:999
3     i1=mod(i,10);
4     i2=mod(floor(i/10),10);
5     i3=floor(i/100);
6     x=i1^3+i2^3+i3^3;
7     if x==i
8         printf('%i\n',x)
9     endif
10 endfor
11
12 #Solution2
13 printf('\n')
14 for i3=1:9
15     for i2=0:9
16         for i1=0:9
17             i=i3*100+i2*10+i1;
18             y=i1^3+i2^3+i3^3;
19             if i==y
20                 printf('%i\n',y)
21             endif
22         endfor
23     endfor
24 endfor
25
```

Output

Output numbers are 153, 370, 371, 407. Since my source code includes two solutions (separated by `printf("\n')`) numbers are written two times in total.

