**6 kyu**

**Deficiently Abundant Perfect Amicable Numbers**

10096% of 2668 of156[RichAWarren](https://www.codewars.com/users/RichAWarren)

Python

* [TRAIN AGAIN](https://www.codewars.com/kata/deficiently-abundant-perfect-amicable-numbers/train/python)
* [NEXT KATA](https://www.codewars.com/trainer/python)

Details

[Solutions](https://www.codewars.com/kata/deficiently-abundant-perfect-amicable-numbers/solutions/python)

[Discourse (3)](https://www.codewars.com/kata/deficiently-abundant-perfect-amicable-numbers/discuss/python)

* Add to Collection
* |
* Share this kata:

For a given two numbers your mission is to derive a function that evaluates whether two given numbers are **abundant**, **deficient** or **perfect** and whether together they are **amicable**.

**Abundant Numbers**

An abundant number or excessive number is a number for which the sum of its proper divisors is greater than the number itself.

The integer 12 is the first abundant number. Its proper divisors are 1, 2, 3, 4 and 6 for a total of 16 (> 12).

**Deficient Numbers**

A deficient number is a number for which the sum of its proper divisors is less than the number itself.

The first few deficient numbers are: 1, 2, 3, 4, 5, 7, 8, 9.

**Perfect Numbers**

A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.

The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6.

**Amicable Numbers**

Amicable numbers are two different numbers so related that the sum of the proper divisors of each is equal to the other number. (A proper divisor of a number is a positive factor of that number other than the number itself. For example, the proper divisors of 6 are 1, 2, and 3.)

For example, the smallest pair of amicable numbers is (220, 284); for the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110, of which the sum is 284; and the proper divisors of 284 are 1, 2, 4, 71 and 142, of which the sum is 220.

**The Function**

For a given two numbers, derive function deficientlyAbundantAmicableNumbers(num1, num2)which returns a string with first and second word either abundant or deficient depending on whether num1 or num2 are abundant, deficient or perfect. The string should finish with either amicable or not amicable depending on the relationship between num1 and num2.

e.g. deficientlyAbundantAmicableNumbers(220, 284) returns "abundant deficient amicable" as 220 is an abundant number, 284 is a deficient number and amicable because 220 and 284 are an amicable number pair.

See Part 1 - [Excessively Abundant Numbers](http://www.codewars.com/kata/56a75b91688b49ad94000015" \t "_blank)

See Part 2 - [The Most Amicable of Numbers](http://www.codewars.com/kata/56b5ebaa26fd54188b000018" \t "_blank)

<https://www.codewars.com/kata/deficiently-abundant-perfect-amicable-numbers/python>

**def** SumDivisores( n):

    s = 0

    i = 1

**while**(i\*i <= n):

**if**(n%i==0):

            s +=i

**if**(n / i != i):

**if**(n/i != n):

                    s += (n/i)

        i+=1

**return** s

**def** deficiently\_abundant\_amicable\_numbers(n1,n2):

    sumDivN1 = SumDivisores(n1)

    sumDivN2 = SumDivisores(n2)

    a = ""

**if**(sumDivN1 == n1):

        a = "perfect"

**elif**(sumDivN1 > n1 ):

        a = "abundant"

**elif** (sumDivN1 < n1):

        a = "deficient"

    b = ""

**if**(sumDivN2 == n2):

        b = "perfect"

**elif**(sumDivN2 > n2):

        b = "abundant"

**elif** (sumDivN2 < n2):

        b = "deficient"

    c = ""

**if**(sumDivN1 == n2 **and** sumDivN2 == n1 ):

        c = "amicable"

**else** :

        c = "not amicable"

**return** a + " " + b + " " + c