Ugly Number

Write a program to check whether a given number is an ugly number.

Ugly numbers are positive numbers whose prime factors only include `2, 3, 5`. For example, `6, 8` are ugly while `14` is not ugly since it includes another prime factor `7`.

Note that `1` is typically treated as an ugly number.

Credits:
Special thanks to @jianchao.li.fighter for adding this problem and creating all test cases.

Solution 1

Just divide by 2, 3 and 5 as often as possible and then check whether we arrived at 1. Also try divisor 4 if that makes the code simpler / less repetitive.

C++ / C

```
for (int i=2; i<6 && num; i++)
    while (num % i == 0)
        num /= i;
return num == 1;
```

Ruby

```
(2..5).each { |i| num /= i while num % i == 0 } if num > 0
num == 1
```

Or:

```
require 'prime'
num > 0 && num.prime_division.all? { |p, _| p < 6 }
```

Python

```
for p in 2, 3, 5:
    while num % p == 0 < num:
        num /= p
return num == 1
```

Java / C#

```
for (int i=2; i<6 && num>0; i++)
    while (num % i == 0)
        num /= i;
return num == 1;
```

Javascript

```
for (var p of [2, 3, 5])
    while (num && num % p == 0)
        num /= p;
return num == 1;
```

General

Would be a bit cleaner if I did the zero-test outside, and discarding negative numbers right away can speed things up a little, but meh... I don't want to add another line and indentation level :-)

```
if (num > 0)
    for (int i=2; i<6; i++)
        while (num % i == 0)
            num /= i;
return num == 1;
```

written by StefanPochmann original link here

Solution 2

```
public boolean isUgly(int num) {
    if(num==1) return true;
    if(num==0) return false;
    while(num%2==0) num=num>>1;
    while(num%3==0) num=num/3;
    while(num%5==0) num=num/5;
    return num==1;
}
```

written by kittyfeng original link here

Solution 3

```java
public boolean isUgly(int num) {
    if (num <= 0) {return false;}
    if (num == 1) {return true;}
    if (num % 2 == 0) {
        return isUgly(num/2);
    }
    if (num % 3 == 0) {
        return isUgly(num/3);
    }
    if (num % 5 == 0) {
        return isUgly(num/5);
    }
    return false;
}
```

---

idea:

- (1) basic cases: <= 0 and == 1
- (2) other cases: since the number can contain the factors of 2, 3, 5, I just remove those factors. So now, I have a number without any factors of 2, 3, 5.
- (3) after the removing, the number (new number) can contain a) the factor that is prime and meanwhile it is >= 7, or b) the factor that is not the prime and the factor is not comprised of 2, 3 or 5. In both cases, it is false (not ugly number).

For example, new number can be 11, 23 --> not ugly number (case a)). new number also can be 49, 121 --> not ugly number (case b))

written by hello_today_ original link here

From Leetcoder.