1. Given a signed 32-bit integer x, return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range [-2^31, 2^31 - 1], then return 0.

Example 1:
Input: x = 123
Output: 321

Example 2:
Input: x = -123
Output: -321

Example 3:
Input: x = 120
Output: 21

Example 4:
Input: x = 0
Output: 0

2. Given an integer n, return true if n is an ugly number. Ugly number is a positive number whose prime factors only include 2, 3, and/or 5.

Example 1:
Input: n = 6
Output: true
Explanation: 6 = 2 × 3

Example 2:
Input: n = 8
Output: true
Explanation: 8 = 2 × 2 × 2

Example 3:
Input: n = 14
Output: false
Explanation: 14 is not ugly since it includes another prime factor 7.

Example 4:
Input: n = 1
Output: true
Explanation: 1 is typically treated as an ugly number.

3. Given an integer array nums, find three numbers whose product is maximum and return the maximum product.

Example 1:
Input: nums = [1,2,3]
Output: 6

Example 2:
Input: nums = [1,2,3,4]
Output: 24

Example 3:
Input: nums = [-1,-2,-3]
Output: -6

4. Given an integer n, return true if it is a power of two. Otherwise, return false.An integer n is a power of two, if there exists an integer x such that n == 2^x.

Example 1:
Input: n = 1
Output: true
Explanation: 20 = 1

Example 2:
Input: n = 16
Output: true
Explanation: 24 = 16

Example 3:
Input: n = 3
Output: false

Example 4:
Input: n = 4
Output: true

Example 5:
Input: n = 5
Output: false

5. Implement a method to perform basic string compression using the counts of repeated characters. For example, the string aabcccccaaa would become a2blc5a3. If the "compressed" string would not become smaller than the original string, your method should return the original string. You can assume the string has only uppercase and lowercase letters (a - z).

Example 1:
Input: "aabcccccaaa"
Output: "a2b1c5a3"

Example 2:
Input: "abbccd"
Output: "abbccd"
Explanation:
The compressed string is "a1b2c2d1", which is longer than the original string.

6. Given a non-negative integer x, compute and return the square root of x.Since the return type is an integer, the decimal digits are truncated, and only the integer part of the result is returned.

Example 1:
Input: x = 4
Output: 2

Example 2:
Input: x = 8
Output: 2
Explanation: The square root of 8 is 2.82842..., and since the decimal part is truncated, 2 is returned.

7. Write an algorithm to determine if a number n is happy. A happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1.Those numbers for which this process ends in 1 are happy. Return true if n is a happy number, and false if not.

Example 1:
Input: n = 19
Output: true
Explanation:
1^2 + 9^2 = 82
8^2 + 2^2 = 68
6^2 + 8^2 = 100
1^2 + 0^2 + 0^2 = 1

Example 2:
Input: n = 2
Output: false