1.  – множество всех четных чисел.

Перечислимое: да, множество всех чётных чисел может быть перечислено алгоритмом: начать с 0 и последовательно добавлять 2.

Разрешимое: да. Для любого числа x, проверить, является ли x чётным, можно за конечное время с помощью операции x%2=0.

1.  – множество всех простых чисел.

Перечислимое: да. Алгоритм, реализующий решето Эратосфена, перечисляет все простые числа.

Разрешимое: да. Существует алгоритм (проверка делимости до sqrt{x}), который определяет за конечное время, простое ли число x.

1.  – множество всех положительных действительных чисел;

Перечислимое: нет. Положительные действительные числа не могут быть перечислены, так как множество всех действительных чисел не счётное.

Разрешимое: нет. Определить принадлежность числа к множеству положительных действительных невозможно за конечное время, так как представление числа может быть бесконечным.

1.  – множество, содержащее натуральные числа 𝑥, 𝑦, 𝑧 для которых ,  – натуральное.

Перечислимое: нет. Согласно теореме Ферма, такие числа не существуют для n>2. Следовательно, множество пустое, а пустое множество, будучи подмножеством счётного множества, не перечислимо.

Разрешимое: да. Для любой заданной тройки (x, y, z) и фиксированного n>2, можно проверить, выполняется ли равенство x^n + y^n = z^n, что всегда даст отрицательный результат.

1.  – множество, содержащее натуральные числа 𝑥, 𝑦, 𝑧 для которых , натуральное .

Перечислимое: да. Это множество содержит все натуральные тройки (x, y, z), так как по теореме Ферма равенство x^n + y^n = z^n не выполняется при n>2n > 2n>2. Перебор всех натуральных троек (x, y, z) позволяет перечислить такие элементы.

Разрешимое: да. Для любой заданной тройки (x, y, z) можно проверить, что x^n + y^n ≠ z^n, так как это будет выполняться всегда для n>2n > 2n>2.

1.  – множество псевдослучайных чисел в диапазоне , сформированных программой.

Перечислимое: да. Программа может последовательно генерировать псевдослучайные числа.

Разрешимое: нет. Потому что мы не можем гарантированно проверить за конечное время, входит ли конкретное число в эту последовательность.

1.  – множество всех псевдослучайных чисел в диапазоне , сформированных программой.

Перечислимое: да. Как и в предыдущем случае, программа может перечислить эти числа.

Разрешимое: нет. Причина аналогична предыдущему случаю.

1.  – множество всех совершенных чисел. Совершенные числа – это такие, сумма всех делителей которых равна самому числу. Например, число 6.

Перечислимое: да. Алгоритм может вычислять сумму делителей каждого числа и проверять, является ли она равной самому числу.

Разрешимое: да. Для любого числа x, можно проверить, является ли оно совершенным.

1.  – множество всех слов, кодирующих машины Тьюринга в фиксированном алфавите.

Перечислимое: да. Машины Тьюринга с фиксированным алфавитом могут быть закодированы и перечислены.

Разрешимое: нет. Задача проверки кода машины Тьюринга на корректность неразрешима.

1.  – множество кодов машин Тьюринга, допускающих все входы, которые являются палиндромами (возможно, наряду с другими входами).

Перечислимое: да. Все такие машины Тьюринга можно перечислить.

Разрешимое: нет. Определение принадлежности к множеству машин, допускающих палиндромы, сводится к проблеме остановки.

1.  – множество всех кодов МТ, которые никогда не совершают сдвиг влево.

Перечислимое: да. Коды таких машин можно перечислить.

Разрешимое: да. Для заданной машины Тьюринга можно проверить, есть ли среди её команд команды сдвига влево.

1.  – язык кодов МТ, которые, начиная с пустой ленты, в конце концов  
   записывают где-либо на ней символ 1.

Перечислимое: да. МТ можно перечислить.

Разрешимое: нет. Это сводится к проблеме остановки.

1. – множество кодов МТ *M*, которые, имея в начальный момент пустую ленту, в конце концов записывают на ней некоторый непустой символ.  
   *Указание*. Если *M* имеет *m* состояний, рассмотрите первые *m* + 1 совершаемых ею переходов.

Перечислимое: да. Если в процессе симуляции M записывает хотя бы один непустой символ на ленту, мы добавляем её код в список.

Разрешимое: нет. если машина M никогда не записывает ничего на ленту (или зацикливается), нам придётся ждать бесконечно долго, чтобы убедиться, что она этого не сделает.