Problem 2979: Most Expensive Item That Can Not Be Bought

Problem Information

Difficulty: Medium
Acceptance Rate: 80.20%
Paid Only: Yes
Tags: Math, Dynamic Programming, Number Theory

Problem Description

You are given two **distinct** **prime** numbers `primeOne` and `primeTwo`.

Alice and Bob are visiting a market. The market has an **infinite** number of items, for **any** positive integer `x` there exists an item whose price is `x`. Alice wants to buy some items from the market to gift to Bob. She has an **infinite** number of coins in the denomination `primeOne` and `primeTwo`. She wants to know the **most expensive** item she can **not** buy to gift to Bob.

Return _the price of the**most expensive** item which Alice can not gift to Bob_.

Example 1:

Input: primeOne = 2, primeTwo = 5 **Output:** 3 **Explanation:** The prices of items which cannot be bought are [1,3]. It can be shown that all items with a price greater than 3 can be bought using a combination of coins of denominations 2 and 5.

Example 2:

Input: primeOne = 5, primeTwo = 7 **Output:** 23 **Explanation:** The prices of items which cannot be bought are [1,2,3,4,6,8,9,11,13,16,18,23]. It can be shown that all items with a price greater than 23 can be bought.

Constraints:

* `1 < primeOne, primeTwo < 104` * `primeOne`, `primeTwo` are prime numbers. * `primeOne * primeTwo < 105`

Code Snippets

C++:

```
class Solution {
public:
int mostExpensiveItem(int primeOne, int primeTwo) {

}
};
```

Java:

```
class Solution {
public int mostExpensiveItem(int primeOne, int primeTwo) {

}
}
```

Python3:

```
class Solution:
def mostExpensiveItem(self, primeOne: int, primeTwo: int) -> int:
```