**Tutorial-6**

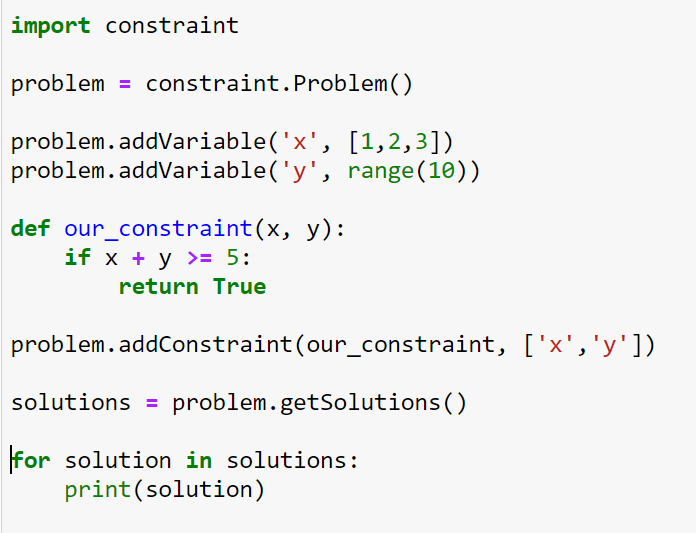
**Constraint Programming**

**Installing the python-constraint Module**

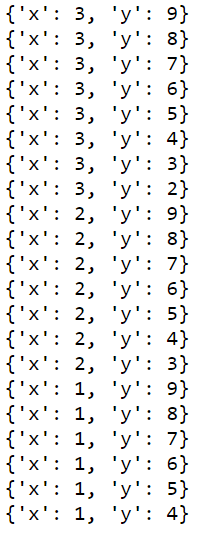
pip install python-constraint

**Example 1**:

**Find all (x,y) where x ∈ {1,2,3} and 0 <= y < 10, and x + y >= 5.**  x is "constrained" to the values 1,2,3, y must be less than 10 and their sum has to be greater than or equal to 5.

**Program:** 

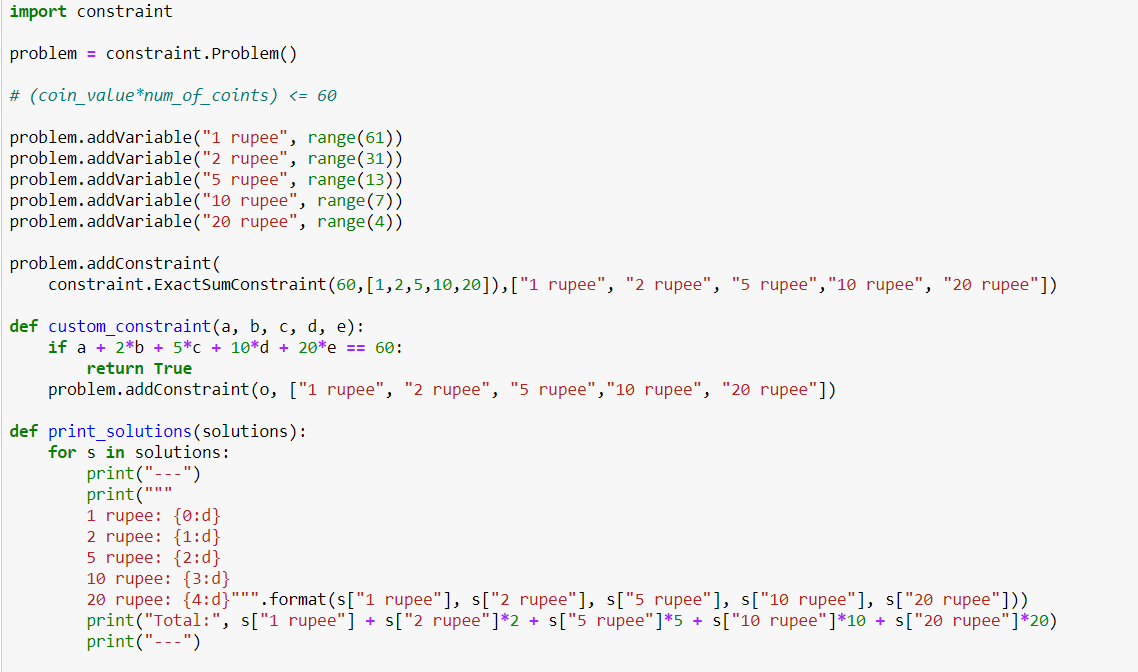
**Output:**

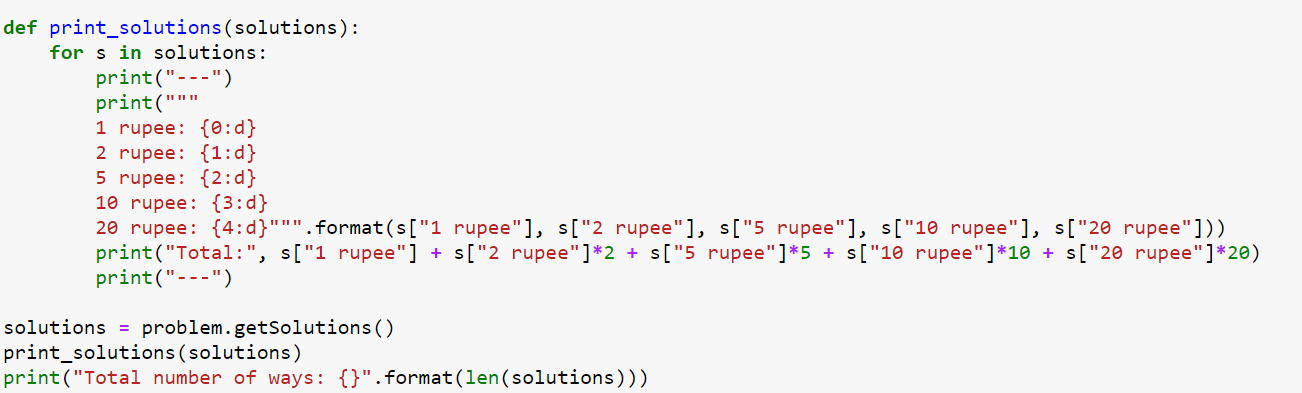


**Example: 2**

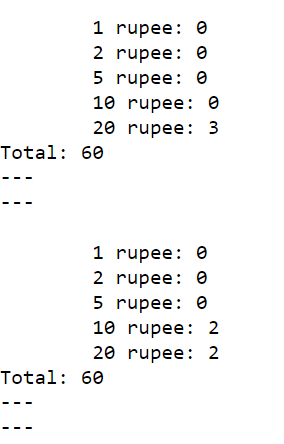
Imagine as you recently got a job as a cashier. You're trying to convince your friend that it's hard work; there are just SO many ways to give someone their change back! Your "friend" shakes his head, obviously not believing you. He says "It can't be that bad. How many ways can there POSSIBLY be to give someone their change back, for like 60 rupees?” Your response is, of course, to sit and quickly write a program that would prove your point. You have a decent amount of coins (2 rupees), (5 rupees), (10 rupees) and (20 rupees), and a lot of coins worth 1 rupee each. Calculate in how many ways you can return change for 60 rupees.

**Program:**





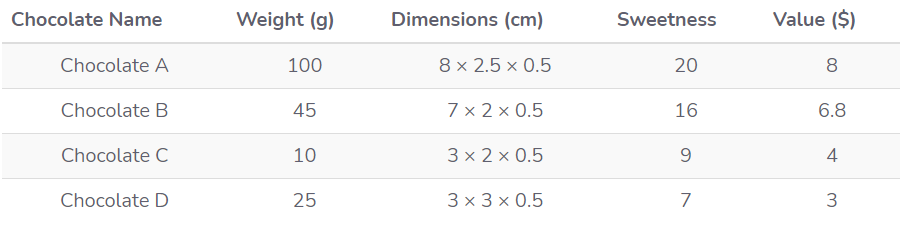
**Output:**



Total number of ways: 782

**Example: 3**

You wish to pack chocolates for your mother. Luckily you work in a chocolate factory that has a lot of leftover chocolate. You have a few chocolate types at your disposal. Your goal is to bring her the sweetest chocolate possible, that you can pack in your bag and sneak through security, and that wouldn't pass a certain net value for which you'd go to prison if you got caught. Security most likely won't get suspicious if you bring less than 3kg. You can fit 1 dm^3 of chocolate in your bag. You won't go to jail if you steal less than $300 worth of chocolate.



We'll first figure out how much of each chocolate we can have if we ONLY brought that type, so we can have the upper bound of our intervals.

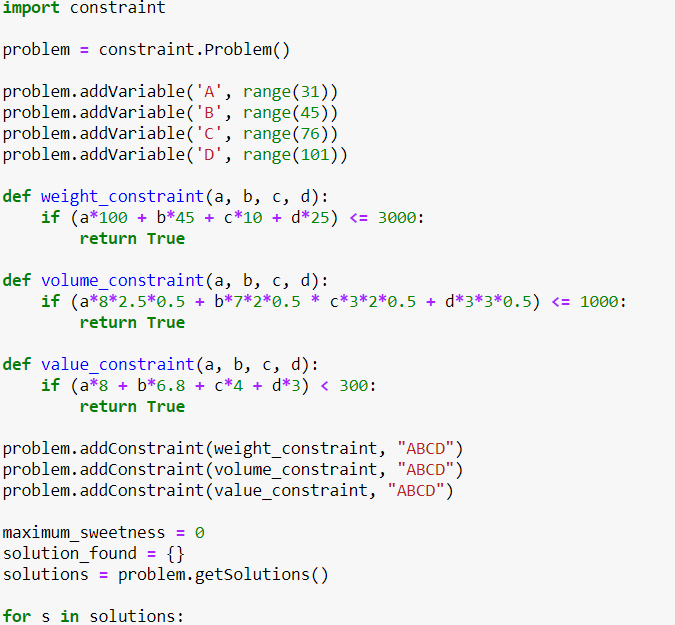
For example, for Chocolate A, based on weight we can bring at most **30** bars, based on value we can bring **37** at most, and based on volume we can bring **100**. The smallest of these numbers is **30**.

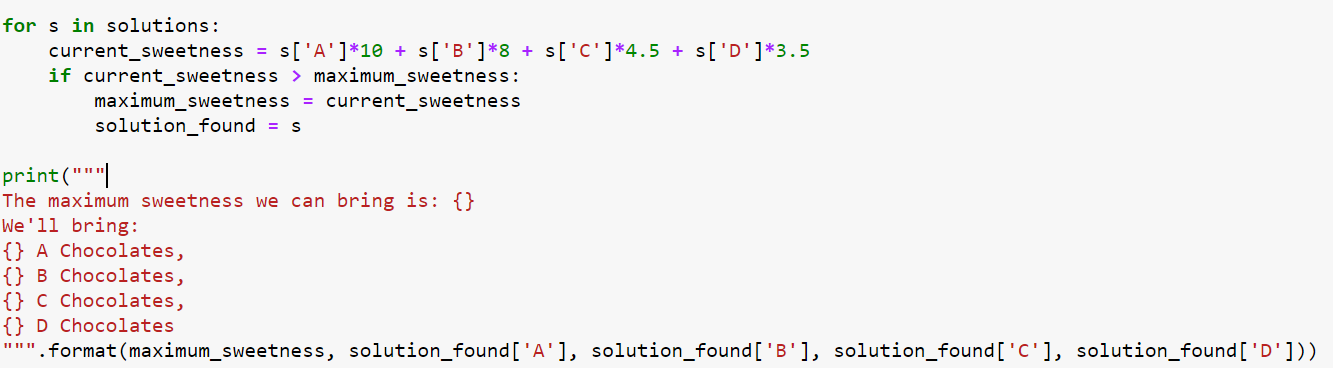
The same steps give us the maximum amount of the rest, **B -> 44, C -> 75, D -> 100**.

3kg = 3,000g

1dm^3 = 1,000cm

**Program:**





**Output:**

