

COP 3223 Program #3: Counting Pez

Part A: Maximum Possible Pez (maxpez.py)

Arup is embarking on an ambitious project: to fill ALL of his Pez dispensers. Normally, this would amount to just counting the number of dispensers and multiplying by 12, the maximum possible candies per dispenser. Unfortunately, Arup has collected some unique dispensers, some of which have different holding capacities of Pez! Write a program to quantify Arup's Pez needs.

When your program begins, prompt the user to enter the number of types of Pez dispensers he/she has. You are guaranteed that this will be a positive integer less than 10. Then, for each type of dispenser, ask the following questions:

- 1) How many Pez fit in one dispenser of this type?
- 2) How many dispensers of this type do you own?

You are guaranteed that the answers to both of these questions will be positive integers less than 1000. At the end of the program, print out the total number of Pez needed to fill the dispensers.

Sample Program Run #1

```
How many types of Pez do you have?  
2  
How many Pez fit in dispenser type #1?  
10  
How many dispensers of type #1 do you have?  
4  
How many Pez fit in dispenser type #2?  
12  
How many dispensers of type #2 do you have?  
5  
You need 100 Pez candies.
```

Sample Program Run #2

```
How many types of Pez do you have?  
3  
How many Pez fit in dispenser type #1?  
1  
How many dispensers of type #1 do you have?  
4  
How many Pez fit in dispenser type #2?  
100  
How many dispensers of type #2 do you have?  
6  
How many Pez fit in dispenser type #3?  
20  
How many dispensers of type #3 do you have?  
4  
You need 684 Pez candies.
```

Part B: Multiplying Pez (multpez.py)

It turns out that at the Pez manufacturing plant in Orange, Connecticut, the company has produced a super-secret type of Pez that procreates! Unfortunately, these new products have yet to be FDA approved, which is why you haven't seen them in stores yet. Because you are so excited about the new upcoming product, you decide that you want to write a program that can accurately calculate how many of these procreating Pez you'll have after a few days of buying an original collection.

In particular, each type of these Pez is given a "reproduction factor", R , which is a positive integer less than 10, as well as a period length, D , which is a positive integer number of days. Each of the current Pez pairs up with another Pez and produces R new Pez every D days. Of course, with all of these Pez running around, you must eat some of them. Let E be the number of Pez you eat every D days. Given these three factors, R , D and E , you can produce a chart showing the number of Pez around after D days, $2D$ days, and so forth.

Consider the example of Pez with a reproduction factor of 3, with a period length of 10 days, where you eat 5 Pez each 10 days and start with 20 Pez. This chart shows how many Pez are available on days 0, 10, 20 and 30

Day	Number of Pez
0	20
10	45
20	106
30	260

In between day 0 and day 10, there are 10 pair of Pez which produce 30 new Pez, for a total of 50. Of these, five are eaten by day 10 for a total of 45 remaining. From day 10 to 20, 22 pairs of Pez (one Pez is alone unfortunately :() produce 66 new Pez for a total of 111, of which 5 are eaten for a grand total of 106 at the end of day 20. Finally, 53 pairs of Pez in between day 20 and 30 produce 159 new Pez, of which 5 are eaten for a total of 260 Pez left ($106 + 159 - 5$) at the end of day 30.

For your program, prompt the user to enter the initial number of Pez (a positive integer less than 1000), the reproductive factor R (a positive integer less than 10), the period length D (a positive integer less than 100), the number of Pez eaten every D days, and an integer N (a positive integer less than 50), the number of periods for which we want the chart printed, assuming that the first line of the chart represents period 0.

Sample Program Run #1

How many reproducing Pez are you starting with?

20

What is their reproductive factor?

3

How long is a single reproductive cycle in days?

10

How many Pez do you eat every 10 days?

5

How many periods do you want the chart printed?

3

Day	Number of Pez
----	-----
0	20
10	45
20	106
30	260

Note: You don't need to match the spacing of this chart exactly, but try to separate out the two columns with a couple tabs ('\t').

Part C: How Old are the Pez? (oldpez.py)

Not only can Pez dispense candy, but they can talk! One of them has given you a riddle that is too difficult for you to solve on your own, but you realize you can use your Python programming skills to write a program that will discover a solution. The riddle is of the following format: There are three Pez, Annie, Bobby and Carmen. You are given the sum of their ages and the product of their ages, and your goal is to figure out all possibilities for how old each of them is. Luckily, you know that all Pez surviving today are in between 1 and 86, inclusive! (Pez were first created by Eduard Haas III in 1927.)

Prompt the user to enter both the sum and product of the ages of the three Pez. Both values will be positive integers such that all possible solutions for the ages are in between 1 and 86, inclusive.

For each possible age setting, print out a single line with the following format:

Annie = A Bobby = B Carmen = C

where A, B and C are ages respectively, that satisfy the input values given.

Print your solutions in order, with A smallest to A largest, breaking ties by order of B, and breaking those ties by order of C.

Sample Program Run #1

What is the sum of Annie, Bobby and Carmen's ages?

20

What is the product of Annie, Bobby and Carmen's ages?

280

Annie = 5 Bobby = 7 Carmen = 8

Annie = 5 Bobby = 8 Carmen = 7

Annie = 7 Bobby = 5 Carmen = 8

Annie = 7 Bobby = 8 Carmen = 5

Annie = 8 Bobby = 5 Carmen = 7

Annie = 8 Bobby = 7 Carmen = 5

Deliverables

Three source files: *maxpez.py*, for your solution to problem A, *multpez.py* for your solution to problem B, and *oldpez.py* for your solution to problem C. All files are to be submitted over WebCourses.

Restrictions

Although you may use other compilers and coding environments, your program must run in IDLE.

Grading Details

Your programs will be graded upon the following criteria:

- 1) Your correctness
- 2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.