COP 3223 Program #3: Pizza Shack

Program A: We Sell Wings Also!!! (wings.py)

The owner of Pizza Shack has realized that other pizza places increase their profits by selling items other than pizza!!! While a pizza purist considers his blasphemy, the owner of Pizza Shack is no purist; it fact, he's willing to sell wings to make more money!!!

A common trick used by other stores is to sell a greater quantity of wings at a better value. (For examples, 5 wings might cost $7.00 while 10 wings only costs $13.00.) Since our owner isn't so sure about the exact pricing of the wings, she would like for you to produce a program for her that will help her experiment with different wing prices.

In particular, she wants your program to print out charts that have prices for different multiples of 5 wings. All of the pricing schemes will follow a decreasing arithmetic sequence for the price of additional wings. To keep things simple, she simply requests that you print out prices in cents, and not worry about tax.

Thus, your program will take the following inputs:

1) Cost of five wings (in cents)
2) The decrease (in cents) for every addition of five wings added to any single order.
3) The maximum number of wings for any order.

With these inputs, you are to produce a chart where each row shows the number of wings and their cost, separated by a tab ("\t") character.

You are guaranteed that the price list will make sense. Namely, the owner won't type in any values that result in a larger order being cheaper than a smaller order.

Input Specification

1. The cost of five wings (in cents) will be a positive integer in between 300 and 1000, inclusive.
2. The decrease (in cents) for every addition of five wings will be a positive integer in between 5 and 50, inclusive.
3. The maximum number of wings for any order will be a positive multiple of 5 less than or equal to 100. (This number will also be such that the cost for the maximum number of wings on the chart will be the most expensive of any of the possible orders.)

Note: YOU DO NOT HAVE TO CHECK TO SEE IF THESE RESTRICTIONS ARE MET WITH IF STATEMENTS IN YOUR CODE. RATHER, THIS IS A GUARANTEE THAT THE GRADER WILL NOT USE TEST CASES OUTSIDE OF THESE BOUNDS.

Sample Run (User input in bold and italics)

```
What is the cost of five wings, in cents?
700
What is the decrease per additional five wings, in cents?
25
What is the maximum number of wings in an order?
50

Wings     Cost(in cents)
-----     --------------
5         700
10        1375
15        2025
20        2650
25        3250
30        3825
35        4375
40        4900
45        5400
50        5875
```

Program B: Wing Edit (wings2.py)
The Pizza Shack owner has realized that using the arithmetic sequence formula has limited the size of her orders. Namely, eventually if you start subtracting a fixed amount from the price of five wings, you eventually get below 0, which leads to an absurd pricing system. Luckily, you have a brilliant idea you've pitched to her: Put a minimum value for the additional cost of five wings added to any order!

Thus, in the sample shown in the previous program, the 5 wings from 45 to 50 wings cost $4.75, but if we extended the chart and looked at the prices of 140 and 145 wings, we'd see that they were identical ($101.50). Instead, what could happen is to stop using the formula to generate the price of five extra wings when it dips below a certain value. In these cases, simply add this fixed certain value for the cost of those five extra wings.

In this program, take your solution to part A and add a single prompt for input, getting the minimum price, in cents, for an additional five wings.

Input Specification
Same as part A, with the added guarantee:

4. The minimum number of cents for an additional five wings will be a positive integer in between 200 and the cost of the first five wings, inclusive.

Sample Run (User input in bold and italics)
```
What is the cost of five wings, in cents?
795
What is the decrease per additional five wings, in cents?
50
What is the maximum number of wings in an order?
70
What is the minimum cost for an additional five wings, in cents?
295
Wings     Cost(in cents)
-----     --------------
5         795
10        1540
15        2235
20        2880
25        3475
30        4020
35        4515
40        4960
45        5355
50        5700
55        5995
60        6290
65        6585
70        6880
```

Program C: Simulating Sales (sales.py)

The owner of Pizza Shack would like for you to write a computer program that simulates sales at Pizza Shack. Through some careful study, she's figured out that a normal distribution (also known as a Gaussian distribution) best models her sales.*

She would like for you to create a program that allows her to look at some statistics for simulating any number of days' worth of sales. She wants the program to take in the following three inputs:

1) The average value of sales in a day at Pizza Shack (in dollars)
2) The standard deviation of sales in a day at Pizza Shack (in dollars)
3) The number of days of the simulation.

She's curious about the following statistics over the course of all the days:

1) The lowest daily sales total (in dollars).
2) The highest daily sales total (in dollars).
3) The **actual** average of all the daily sales totals for the simulation.
4) The number of days for which sales are one standard deviation above the average that was used to generate the simulation.

The last item will be based on the first two input values. Namely, if the first input value is $1000 and the second is $200, then for the last item, we would want to see the number of days with sales above $1200. Note that if we were to take the **actual** average of these days in the simulation, they might not equal $1000 exactly.

In order to simulate a single sale, you will have to use the following function in the random library:

`random.gauss`(*mu*, *sigma*)¶

Gaussian distribution. *mu* is the mean, and *sigma* is the standard deviation. Returns a random value generated with the specified normal distribution.

You can find the documentation for all the functions in the random library here:

https://docs.python.org/3.0/library/random.html

* - Real restaurant sales probably have different distributions on different days and different times of the year. For example, many probably consistently get more sales on a Friday than a Tuesday. Those around UCF probably do better during the school year. But given that all of these factors are fixed (say we look at Fridays during the Fall semester), we might expect a normal distribution of sales.

Sample Program Run (User Input in Bold and Italics)

```
What is the average value of sales a day?
1000
What is the standard deviation of sales a day?
200
How many days do you want to simulate?
100
The lowest daily sales were $299.5003163090596.
The highest daily sales were $1436.1509911664045.
The average daily sales were $989.9605128695194.
The number of days above $1200.0 was 19.
```

Sample Program Run (User Input in Bold and Italics)

```
What is the average value of sales a day?
1000
What is the standard deviation of sales a day?
200
How many days do you want to simulate?
100
The lowest daily sales were $428.62433377684454.
The highest daily sales were $1490.7184401204745.
The average daily sales were $1004.6493813612119.
The number of days above $1200.0 was 18.
```

Sample Program Run (User Input in Bold and Italics)

```
What is the average value of sales a day?
500
What is the standard deviation of sales a day?
145
How many days do you want to simulate?
50
The lowest daily sales were $121.12031736737066.
The highest daily sales were $849.7255046319253.
The average daily sales were $478.9856685144279.
The number of days above $645.0 was 5.
```

Note: It's near impossible that your program will match this output exactly! Rather, the whole point of simulations is that each time you'll see some differences, but over many trials, answers tend to converge to reasonably predictable ranges. The graders will not judge correctness by looking at the answers only for this problem, but they'll actually analyze your code, in addition to making sure that your answers are in the likely ranges that we expect.

Deliverables
Please submit three separate .py files for your solutions to these problems via WebCourses by the designated due date:

Program A: **wings.py**
Program B: **wings2 .py**
Program C: **sales.py**