CPSC 217 Exercise 4: Everything is on Sale!

Due: Friday November 6 at 12:00 noon

This exercise may be completed individually or as part of a group. If the exercise is completed as a group then all members of the group are expected to make a meaningful contribution to the solution.

Task:

Ben's Bargain Basement has opened a retail location so that customers can view the products available in person. Because it's a bargain basement, everything is on sale! But different kinds of products are discounted by different amounts. To help customers figure out what the reduced price of their purchase will be, Ben wants to be able to print out discount tables. An example of such a table, with 3 discount columns (plus the original price column) and 5 rows, is shown below:

```
original    price discounted by
    price      20%      30%      40%
     5.00     4.00     3.50     3.00
    10.00     8.00     7.00     6.00
    15.00    12.00    10.50     9.00
    20.00    16.00    14.00    12.00
    25.00    20.00    17.50    15.00
```

Write a program that generates such a discount table. The number of discount columns and the number of rows in the table will be entered by the user. The first column in the table will always be the original price. The first discount column in the table will always be for a 20% discount. Any additional columns should be for 30%, 40%, 50%, etc. The largest number of columns that will ever be request by the user is 6, meaning that the right most column would be a 70% discount.

The first row in the table (after the headings) will always be for 5.00 dollars. Each additional row should be for the next multiple of 5.00 dollars, meaning that the second row would be for 10.00 dollars, the third row would be for 15.00 dollars, etc. The user will never request more than 10 rows in the discount table.

Requirements:

- Your program does not need to do any error checking – the user will always provide integer inputs between 1 and the upper limits outlined in the previous paragraphs.
- Your program must use a nested loop to construct the table.
- Your columns of values must line up nicely. While they don't need to have exactly the same spacing as the samples shown on the next page, the spacing should be reasonable and all of the decimal points in each column of the numbers must line up.

Hints:

- You can print out a literal, variable or any other value without moving down to the next line by providing a value to print's end parameter. For example, the statement `print(i, end="")` will display the value stored in the variable `i` without moving down to the next line. If you need to move down to the next line without printing anything else you can use the statement `print()`.

- My program began with 2 lines to read the input. This was followed by 5 lines (including a for loop) to display the column headings. My program concluded with 5 lines (including a nested for loop) to display all of the values in the table. I suggest that your program follow a similar structure, though the exact number of lines in each section might be slightly different.
- You can use the format specifier "%9.2f" to indicate that a number should be formatted using two decimal points with enough leading spaces so that it will occupy a total of 9 columns. Using this format specifier (or something very similar) will help you get your columns lined up nicely.

Sample Run 1:

```
How many columns should there be (1-6)? 1
How many rows should there be in the table? 1

 original    price discounted by
    price      20%
     5.00     4.00
```

Sample Run 2:

```
How many columns should there be (1-6)? 2
How many rows should there be in the table? 10

 original    price discounted by
    price      20%      30%
     5.00     4.00     3.50
    10.00     8.00     7.00
    15.00    12.00    10.50
    20.00    16.00    14.00
    25.00    20.00    17.50
    30.00    24.00    21.00
    35.00    28.00    24.50
    40.00    32.00    28.00
    45.00    36.00    31.50
    50.00    40.00    35.00
```

Sample Run 3:

```
How many columns should there be (1-6)? 6
How many rows should there be in the table? 4

 original    price discounted by
    price      20%      30%      40%      50%      60%      70%
     5.00     4.00     3.50     3.00     2.50     2.00     1.50
    10.00     8.00     7.00     6.00     5.00     4.00     3.00
    15.00    12.00    10.50     9.00     7.50     6.00     4.50
    20.00    16.00    14.00    12.00    10.00     8.00     6.00
```

Grading:

Your program will be graded by testing it with two different sets of input, which may be different from the samples from examples shown above. It will be graded based on whether or not it generates correct values and whether or not the columns are formatted nicely. Your program must use a nested loop to generate its output. A program that does not use a nested loop will receive a grade of F even if all of the output is correct.

Submission Instructions:

Submit your solution as a Python source code file electronically to the Exercise 4 drop box in D2L. You do **not** need to submit a paper copy of your solution. If you choose to complete this exercise as part of group then each member of the group must submit a copy of the exercise using D2L.