

## Directions

You will create a new **YourOhioIdE2.py** file which will include comments and code to address the following problems.

**IMPORTANT:** Like our homework assignments, be sure to adhere to the “coding expectations” document.

For each problem, all numeric data and calculations must be stored in variables. Those variables should be used to within your calculations which will be used to generate the expected output.

Be aware that I will likely not answer questions about the exam. This is all material that you should be familiar with. Read through the problems and address the requirements. Good luck!

You may use your homework, notes, PowerPoint slides, and videos. You may **NOT** use the Internet to search for any python examples, consult with other students, or anyone else, by any means. Do not post your solutions anywhere other than what you submit here. When you are finished, **by submitting your exam solutions, you agree to this.**

## Problems

Here is the dataset that you will use for the problems (*pay close attention to data types – only put quotes around string data when storing it in a list – invoice, quantity, and price are not string data*):

<u>invoice</u>	<u>description</u>	<u>quantity</u>	<u>price</u>	<u>customerId</u>
1	hand warmer	6	1.85	C1
3	red polka dot jar	6	1.85	C2
5	jam making set	6	4.25	C3
7	red coat rack	3	4.95	C4
9	yellow coat rack	3	4.95	C5

1. Create a list that contains all the invoice numbers and display the contents of that list.

**Expected output example:**

```
[1, 3, 5, 7, 9]
```

2. How many invoice numbers do we have? (*programmatically answer the question*)

**Expected output example:**

```
The list contains 5 invoice numbers
```

3. Create separate lists that contain each of the other columns (one for description, quantity, price, and customerId). Display the first value in each of the lists indicating what each item is like you see in the example output (*execute 5 print functions at the same time to do this*).

**Expected output:**

```
Invoice Number: 1
Description: hand warmer
Quantity: 6
Price: 1.85
Customer ID: C1
```

4. Use the **remove()** method of your list variables to remove the first value of all 5 lists. When using the remove method, like the append method, you must reference the data directly (*it does not use an index value*). Once you remove the first value from all 5 lists, print out the new first value of each of the lists like the previous problem:

**Expected output:**

```
Invoice Number: 3
Description: red polka dot jar
Quantity: 6
Price: 1.85
Customer ID: C2
```

5. Based on our dataset, you can logically infer that our next invoice number will be 10. Create a new variable called theNextNewInvoiceNumber. To assign it a value, programmatically find the biggest value in the list of invoice numbers, add 1 to it, and store that value in your new variable. Display the contents of that new variable:

**Expected output:**

```
The next new invoice number will be 10
```

6. Using the **append()** method your list, add your new invoice number to your existing invoice number list variable. In the other 4 list variables append the following data:

```
description: hanger
quantity: 20
price: 5.95
customerId: C2
```

Display the output for the last value of each list.

**Expected output:**

```
Invoice Number: 10
Description: hanger
Quantity: 20
Price: 5.95
Customer ID: C2
```

7. Using your list variable containing price data, display the mean, maximum and minimum values:

**Expected output:**

```
mean: $4.39
maximum: $5.95
minimum: $1.85
```

8. Create a new **set** that will contain a list of unique customerIds. Create the new set variable by converting your existing list of customerIds to a set and store it in the new variable. Display the number of items in the set along with the contents of your set in sorted order.

**Expected output:**

There are 4 customerIds. Here is the list in sorted order: ['C2', 'C3', 'C4', 'C5']

9. Build a dictionary from the 5 lists that you have created where the key is the invoice, and value is a tuple that includes the description, quantity, price, and customerId. Display the keys and values of that dictionary.

**Expected output:**

All keys in the dictionary are:  
`dict_keys([3, 5, 7, 9, 10])`

All values in the dictionary are:  
`dict_values([('red polka dot jar', 6, 1.85, 'C2'), ('jam making set', 6, 4.25, 'C3'), ('red coat rack', 3, 4.95, 'C4'), ('yellow coat rack', 3, 4.95, 'C5'), ('hanger', 20, 5.95, 'C2')])`

10. **popitem()** is a method of the dictionary variable created in the previous problem (e.g., `myDictionary.popitem()`). Unlike using the `keys()` or `values()` methods of your dictionary variable, `popitem` will remove the last key/value pair from your dictionary. Use this method to remove the last 2 key/value pairs from your dictionary (*this means that you will have to run the `popitem` method twice*). Once you do that, display the keys and values of your dictionary.

**Expected output:**

All keys in the dictionary are:  
`dict_keys([3, 5, 7])`

All values in the dictionary are:  
`dict_values([('red polka dot jar', 6, 1.85, 'C2'), ('jam making set', 6, 4.25, 'C3'), ('red coat rack', 3, 4.95, 'C4')])`