"Shopping Cart" Project Checkpoints

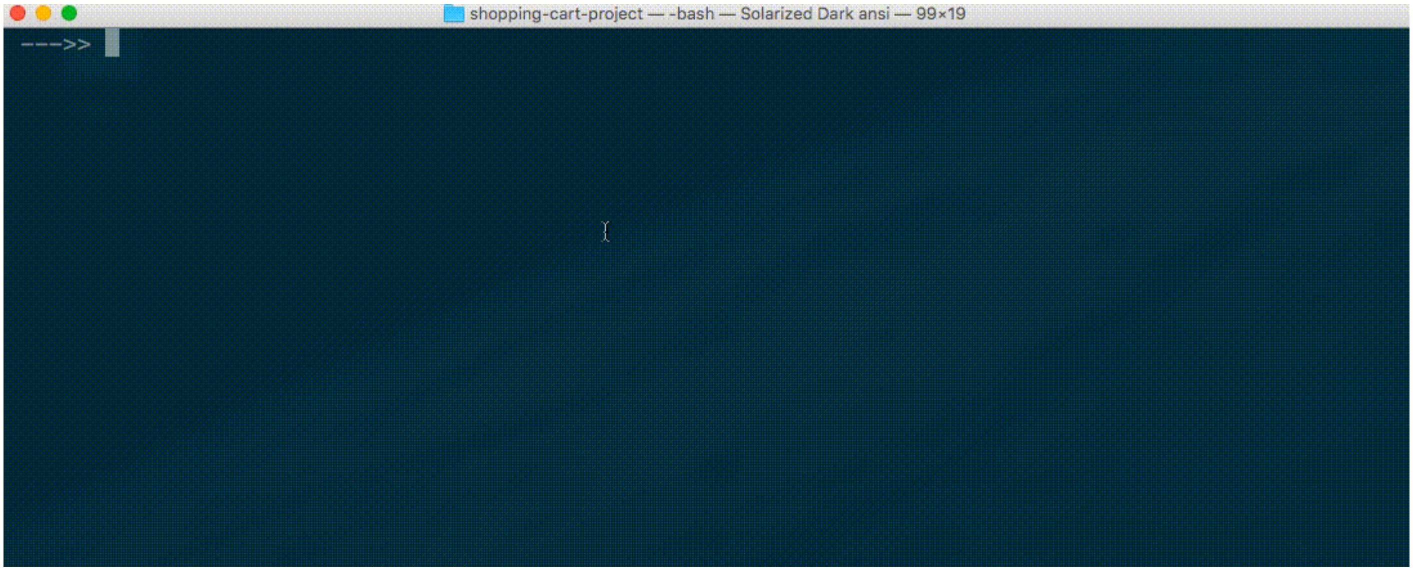
Checkpoint 1: Capturing User Inputs

The user input part of this project is probably the hardest part. This is because it represents something we "know we don't know" how to do. We should try to research and implement this part of the project as soon as possible to relieve our stress and to demonstrate to ourselves it can be done. If you'd rather do the easy parts first, skip this first checkpoint and come back to it after you finish the others.

Steps:

1. Accept a user input value, store it in a variable, and print it. HINT: use the `input()` function 😺.
2. One at a time, iteratively accept a user input value, store it in a variable, and print it. HINT: use an infinite `while` loop. NOTE: you may have to press "control-c" to quit your script if you get stuck.
3. One at a time, iteratively accept a user input value, store it in a variable, and print it. But stop the loop if the user inputs the word "DONE". HINT: use an `if` statement in conjunction with the `break` keyword.
4. Repeat the previous step, but instead of printing each user input, store them all in a single list. Then print the list after the user is "DONE".

When you have finished this checkpoint, your program should perform like this:



```
--->>
```

Checkpoint 2: Look-up Products

If you already did the first checkpoint, great job capturing and storing the user inputs! But now it's time to set that code aside and focus on what will happen after the product identifiers have been captured.

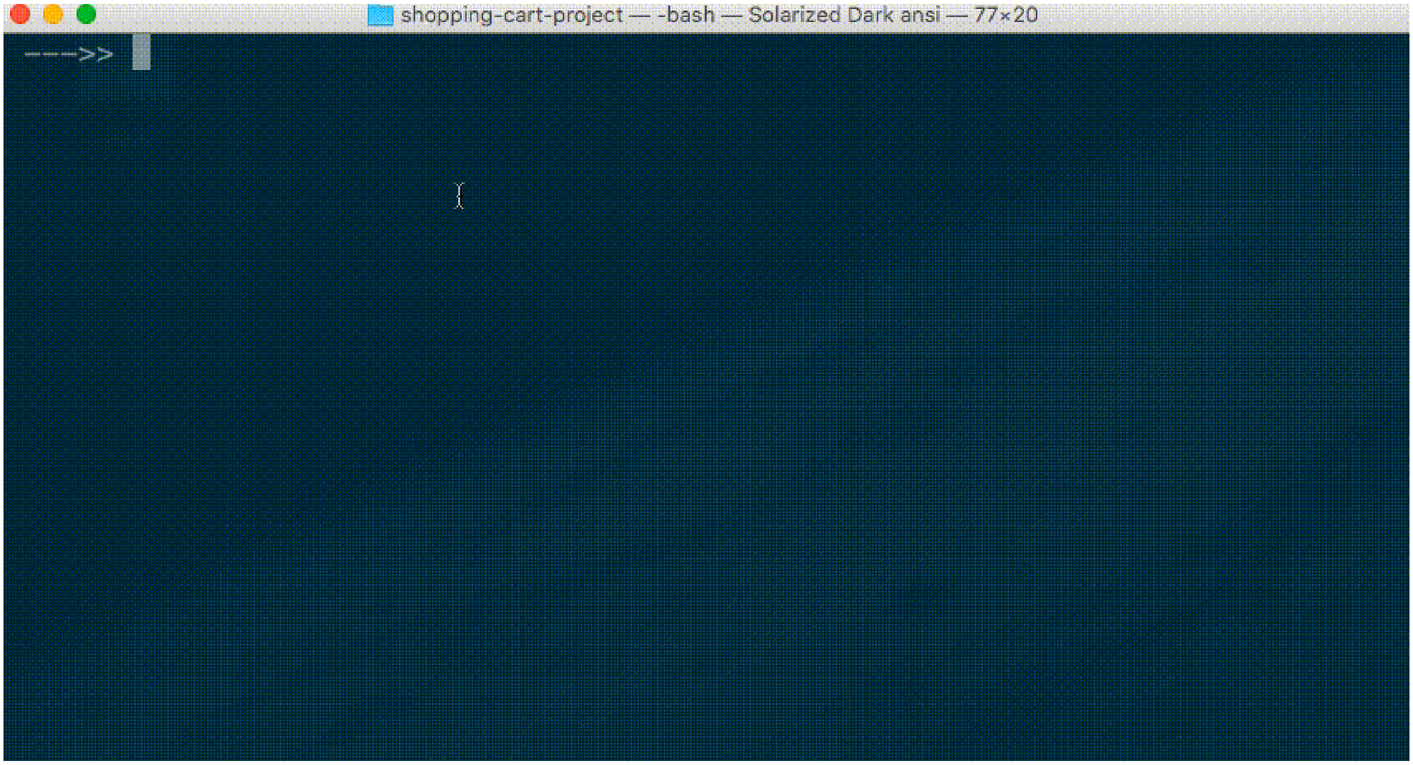
To speed-up the development process, we'll temporarily shift our approach to use a hard-coded list of product identifiers instead of asking for inputs each time we want to test the program. At this time, your script might look something like this:

```python
products = [...] #<--- that long list of product dictionaries provided above

#
# some commented-out loop
# ... representing the result of the first checkpoint (if you did it)
# ... which accepts user inputs
# ... and prints the results
# ... and which we are temporarily ignoring
# ... (yours will actually be some working python code)
#

product_ids = [1, 8, 6, 16, 6] # temporary list of valid ids for testing purposes

print("SHOPPING CART ITEM IDENTIFIERS INCLUDE:", product_ids)

#TODO: perform product look-ups here!
```

Steps:

1. For a single valid product identifier, look up the matching product and print its name and price. HINT: try using a custom function in conjunction with a list comprehension.
2. For each valid product identifier in the example list, look up the matching product and print its name and price.
3. For each valid product identifier in the example list, look up the matching product and print its name and price, and add its price to a running-total of all prices, then print the running-total after iterating through the entire list. For now, you don't necessarily need to worry about formatting prices as USD.

When you have finished this checkpoint, your program should perform something like this:



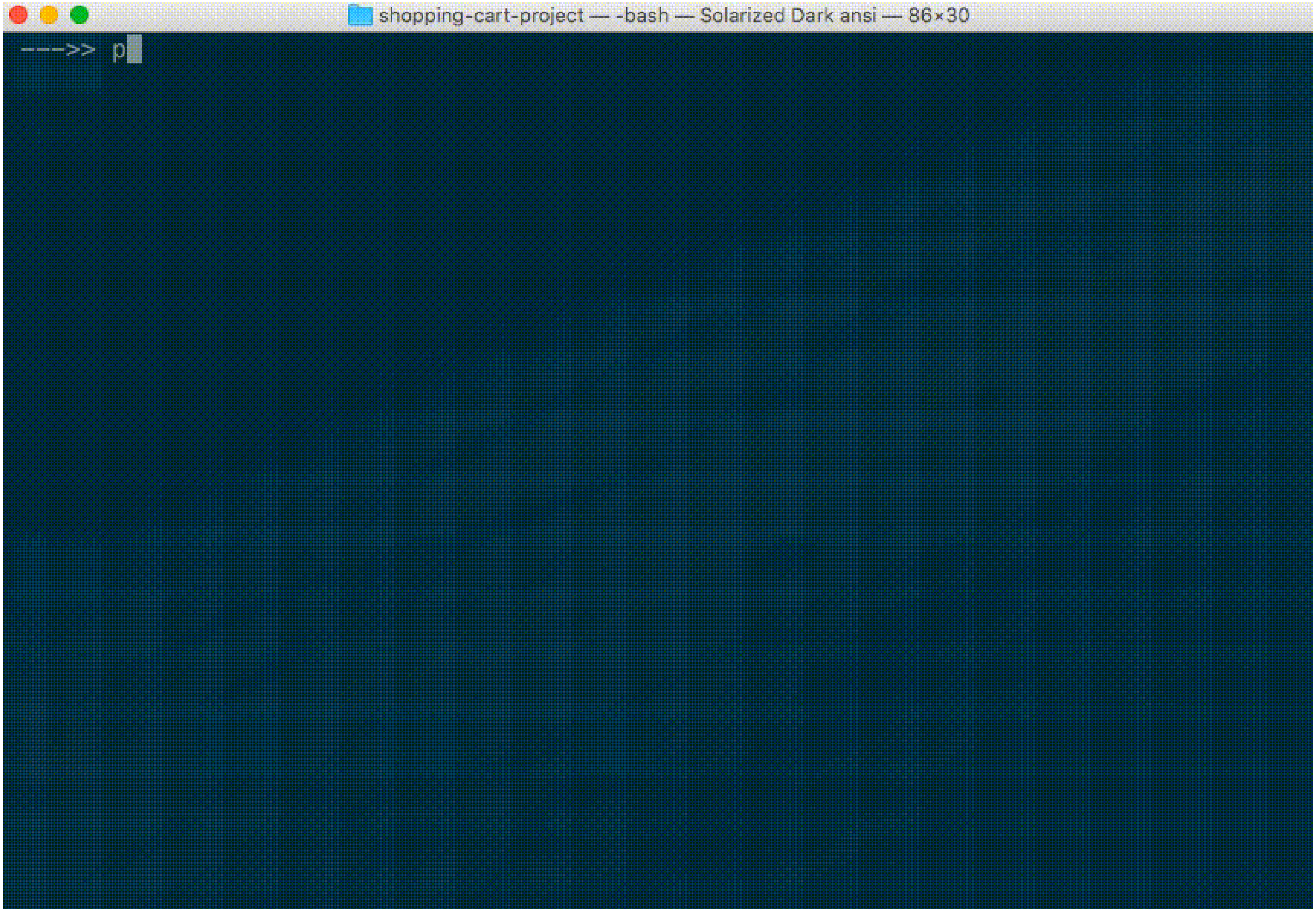
```
--->> 
```

Checkpoint 3: Printing the Receipt

Steps:

1. For each receipt component listed in the project requirements (e.g. store name, product prices, taxes, total price, farewell message, etc.), revise your program to print that component.

When you have finished this checkpoint, your program should perform like this:



```
--->> p
```

Once your program prints all required receipt components, it's time to stop using the hard-coded product identifiers. If you already did the first checkpoint, un-comment the code which performs the user input process, otherwise do the first checkpoint now. Afterwards, revise and configure the other parts of the program as necessary to use the list of product identifiers captured during the user input process.

Wow, you are finally done. Nice job! Now it's time to submit your work...