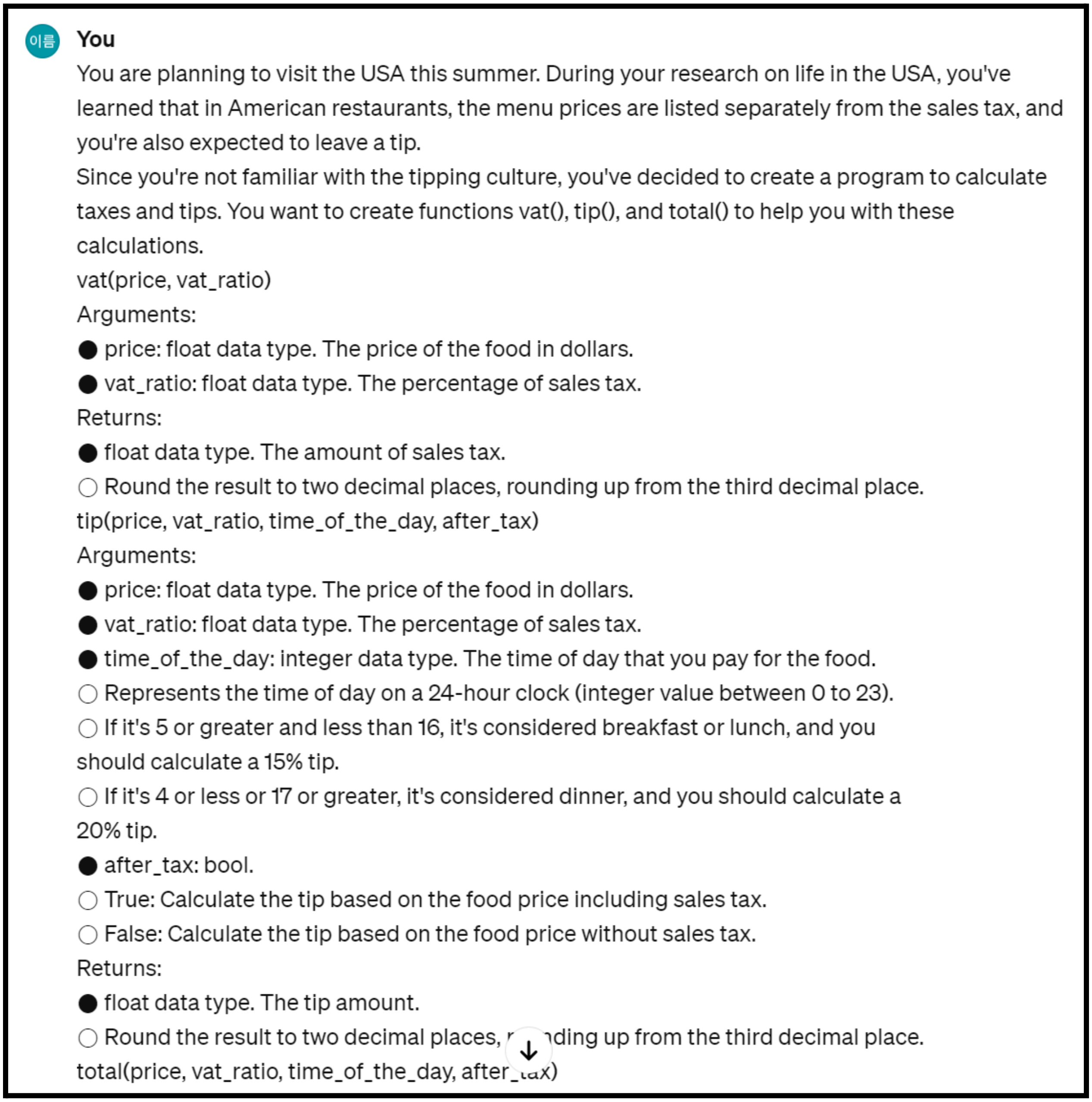
Second Project


You

You are planning to visit the USA this summer. During your research on life in the USA, you've learned that in American restaurants, the menu prices are listed separately from the sales tax, and you're also expected to leave a tip.

Since you're not familiar with the tipping culture, you've decided to create a program to calculate taxes and tips. You want to create functions vat(), tip(), and total() to help you with these calculations.

vat(price, vat_ratio)

Arguments:

- price: float data type. The price of the food in dollars.
- vat_ratio: float data type. The percentage of sales tax.

Returns:

- float data type. The amount of sales tax.
  - Round the result to two decimal places, rounding up from the third decimal place.

tip(price, vat_ratio, time_of_the_day, after_tax)

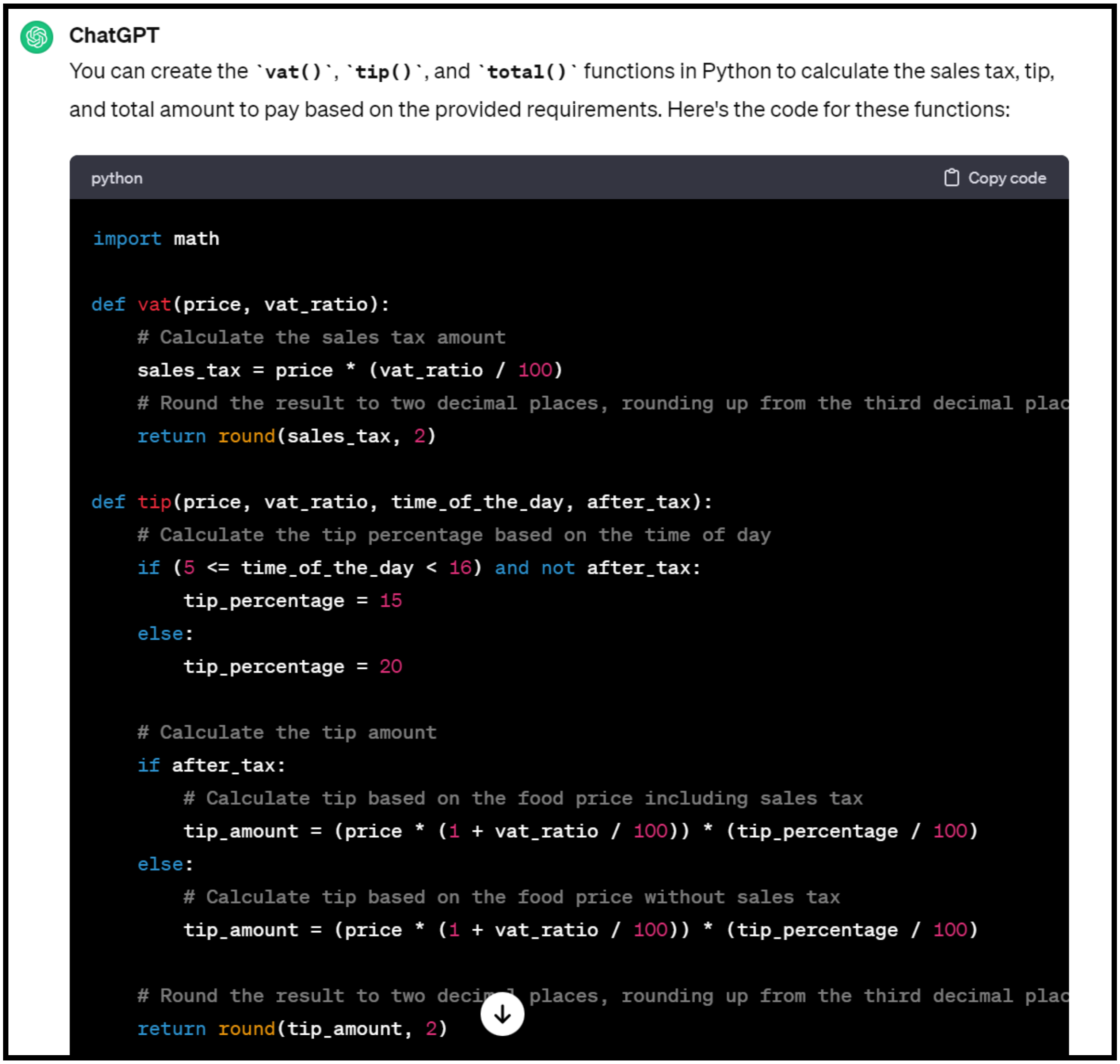
Arguments:

- price: float data type. The price of the food in dollars.
- vat_ratio: float data type. The percentage of sales tax.
- time_of_the_day: integer data type. The time of day that you pay for the food.
  - Represents the time of day on a 24-hour clock (integer value between 0 to 23).
  - If it's 5 or greater and less than 16, it's considered breakfast or lunch, and you should calculate a 15% tip.
  - If it's 4 or less or 17 or greater, it's considered dinner, and you should calculate a 20% tip.
- after_tax: bool.
  - True: Calculate the tip based on the food price including sales tax.
  - False: Calculate the tip based on the food price without sales tax.

Returns:

- float data type. The tip amount.
  - Round the result to two decimal places, rounding up from the third decimal place.

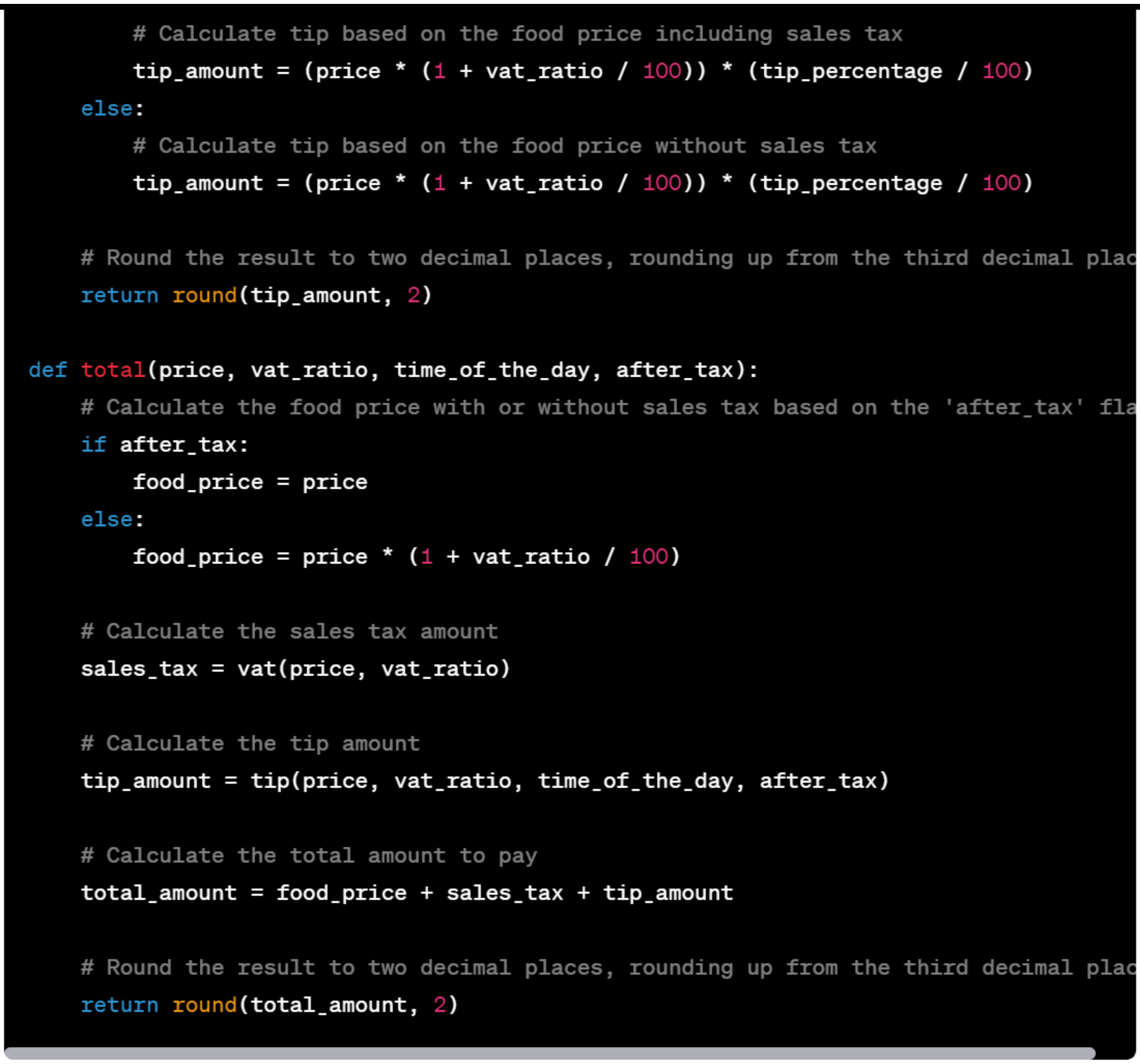
total(price, vat_ratio, time_of_the_day, after_tax)



ChatGPT

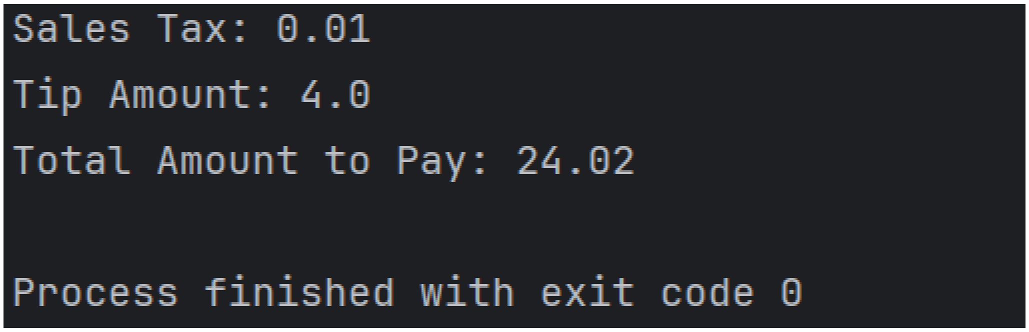
You can create the `vat()`, `tip()`, and `total()` functions in Python to calculate the sales tax, tip, and total amount to pay based on the provided requirements. Here's the code for these functions:

```python
import math

def vat(price, vat_ratio):
    # Calculate the sales tax amount
    sales_tax = price * (vat_ratio / 100)
    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(sales_tax, 2)

def tip(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the tip percentage based on the time of day
    if (5 <= time_of_the_day < 16) and not after_tax:
        tip_percentage = 15
    else:
        tip_percentage = 20

    # Calculate the tip amount
    if after_tax:
        # Calculate tip based on the food price including sales tax
        tip_amount = (price * (1 + vat_ratio / 100)) * (tip_percentage / 100)
    else:
        # Calculate tip based on the food price without sales tax
        tip_amount = (price * (1 + vat_ratio / 100)) * (tip_percentage / 100)

    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(tip_amount, 2)
```



```
        # Calculate tip based on the food price including sales tax
        tip_amount = (price * (1 + vat_ratio / 100)) * (tip_percentage / 100)
    else:
        # Calculate tip based on the food price without sales tax
        tip_amount = (price * (1 + vat_ratio / 100)) * (tip_percentage / 100)

    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(tip_amount, 2)

def total(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the food price with or without sales tax based on the 'after_tax' fla
    if after_tax:
        food_price = price
    else:
        food_price = price * (1 + vat_ratio / 100)

    # Calculate the sales tax amount
    sales_tax = vat(price, vat_ratio)

    # Calculate the tip amount
    tip_amount = tip(price, vat_ratio, time_of_the_day, after_tax)

    # Calculate the total amount to pay
    total_amount = food_price + sales_tax + tip_amount

    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(total_amount, 2)
```

In the first question, I input the whole Table 1 into the prompt. It's more efficient than typing down the whole information of the functions. And then ChatGPT gave me the code. There are three functions and the names of the functions and arguments are all correct.

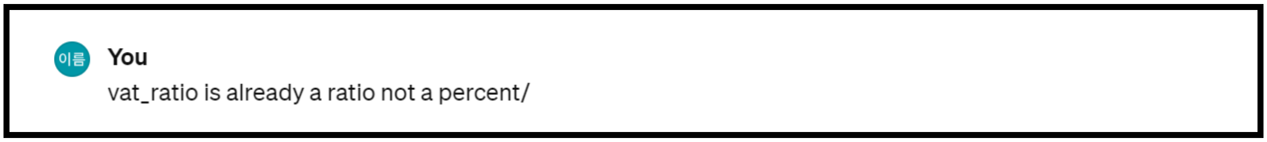


```
Sales Tax: 0.01
Tip Amount: 4.0
Total Amount to Pay: 24.02

Process finished with exit code 0
```

But when I tested the script, I couldn't get the expected output. For instance, when the input is 20, 0.06 the expected output from vat function is 1.2, but the actual output is 0.01.

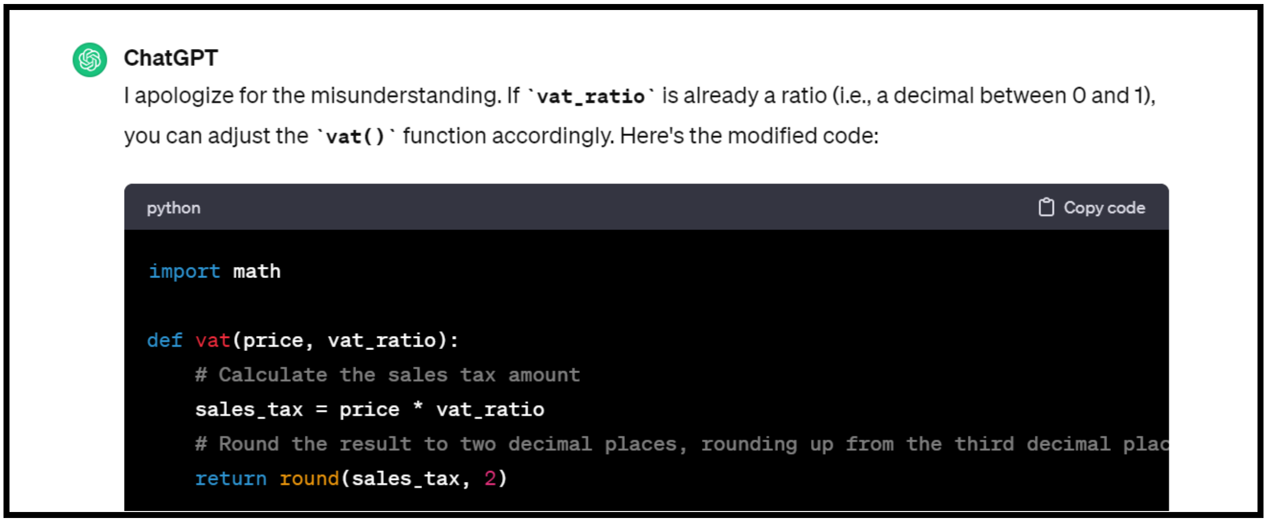
That was too small for the expected one. Based on the gap of the size between two of them, I guessed that vat_ratio, which is less than 1, is considered as a percentage in the script and that was correct.

The reason why I couldn't catch the error at first is that I thought ChatGPT could understand the context from the variable's name. For example, I thought ChatGPT would



You

vat_ratio is already a ratio not a percent/



ChatGPT

I apologize for the misunderstanding. If `vat_ratio` is already a ratio (i.e., a decimal between 0 and 1), you can adjust the `vat()` function accordingly. Here's the modified code:

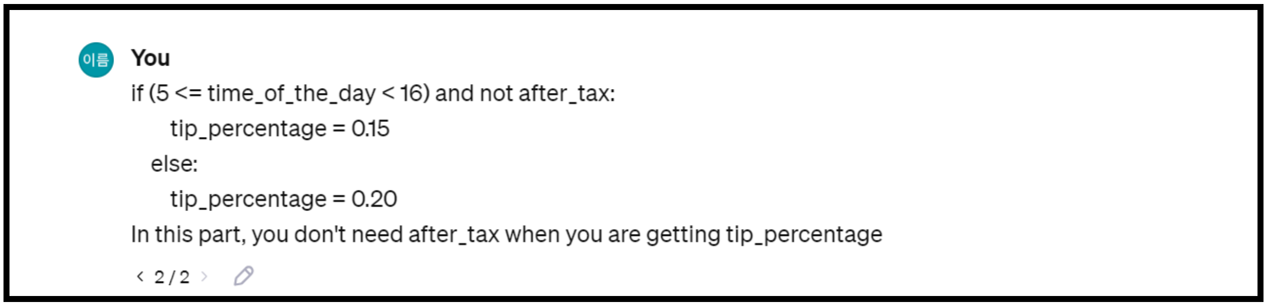
```python
import math

def vat(price, vat_ratio):
    # Calculate the sales tax amount
    sales_tax = price * vat_ratio
    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(sales_tax, 2)
```

So, I indicated to ChatGPT that vat_ratio is not a percentage.

```
sales_tax = price * (vat_ratio / 100)
```

In turn, ChatGPT modified the script by not dividing vat_ratio into 100.

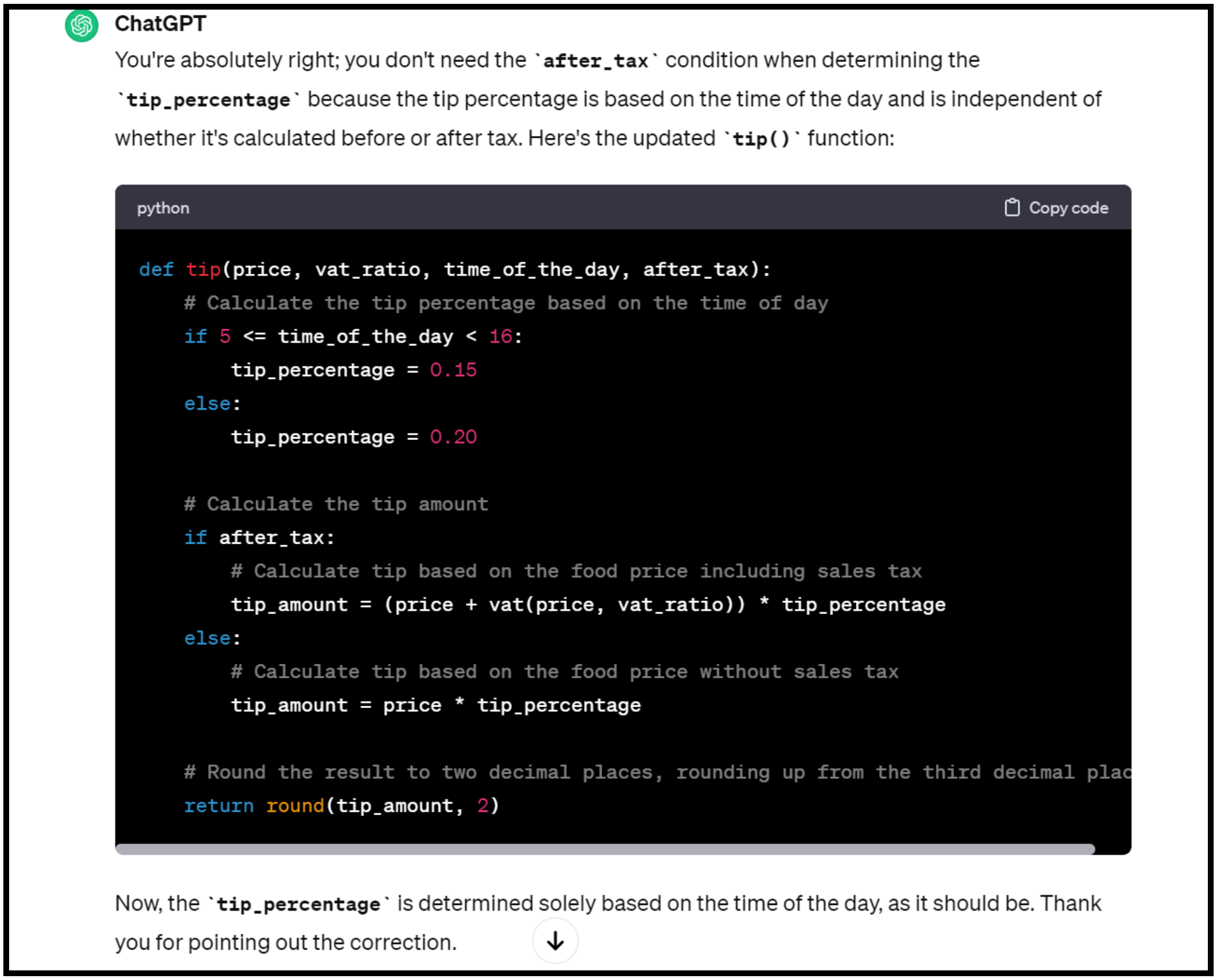
```
sales_tax = price * vat_ratio
```



You

```
if (5 <= time_of_the_day < 16) and not after_tax:
    tip_percentage = 0.15
  else:
    tip_percentage = 0.20
```

In this part, you don't need after_tax when you are getting tip_percentage



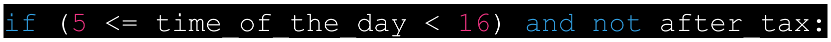
ChatGPT

You're absolutely right; you don't need the `after_tax` condition when determining the `tip_percentage` because the tip percentage is based on the time of the day and is independent of whether it's calculated before or after tax. Here's the updated `tip()` function:

```python
def tip(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the tip percentage based on the time of day
    if 5 <= time_of_the_day < 16:
        tip_percentage = 0.15
    else:
        tip_percentage = 0.20

    # Calculate the tip amount
    if after_tax:
        # Calculate tip based on the food price including sales tax
        tip_amount = (price + vat(price, vat_ratio)) * tip_percentage
    else:
        # Calculate tip based on the food price without sales tax
        tip_amount = price * tip_percentage

    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(tip_amount, 2)
```

Now, the `tip_percentage` is determined solely based on the time of the day, as it should be. Thank you for pointing out the correction.

And the problem was also in the tip function. The problem was in the if condition. The tip_percentage must be determined by only time_of_the_day. The variable 'after_tax' should not interrupt the determination.

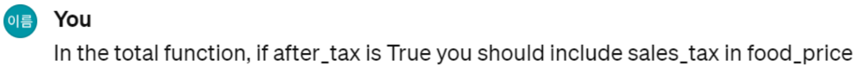


```
if (5 <= time_of_the_day < 16) and not after_tax:
```

I guess, ChatGPT might have an issue with processing boolean type instances.

```
if 5 <= time_of_the_day < 16:
```

In the modified script, after_tax is perfectly separated from the if condition when determining tip_percentage. After this modification, the function works perfectly and I can get the expected output.

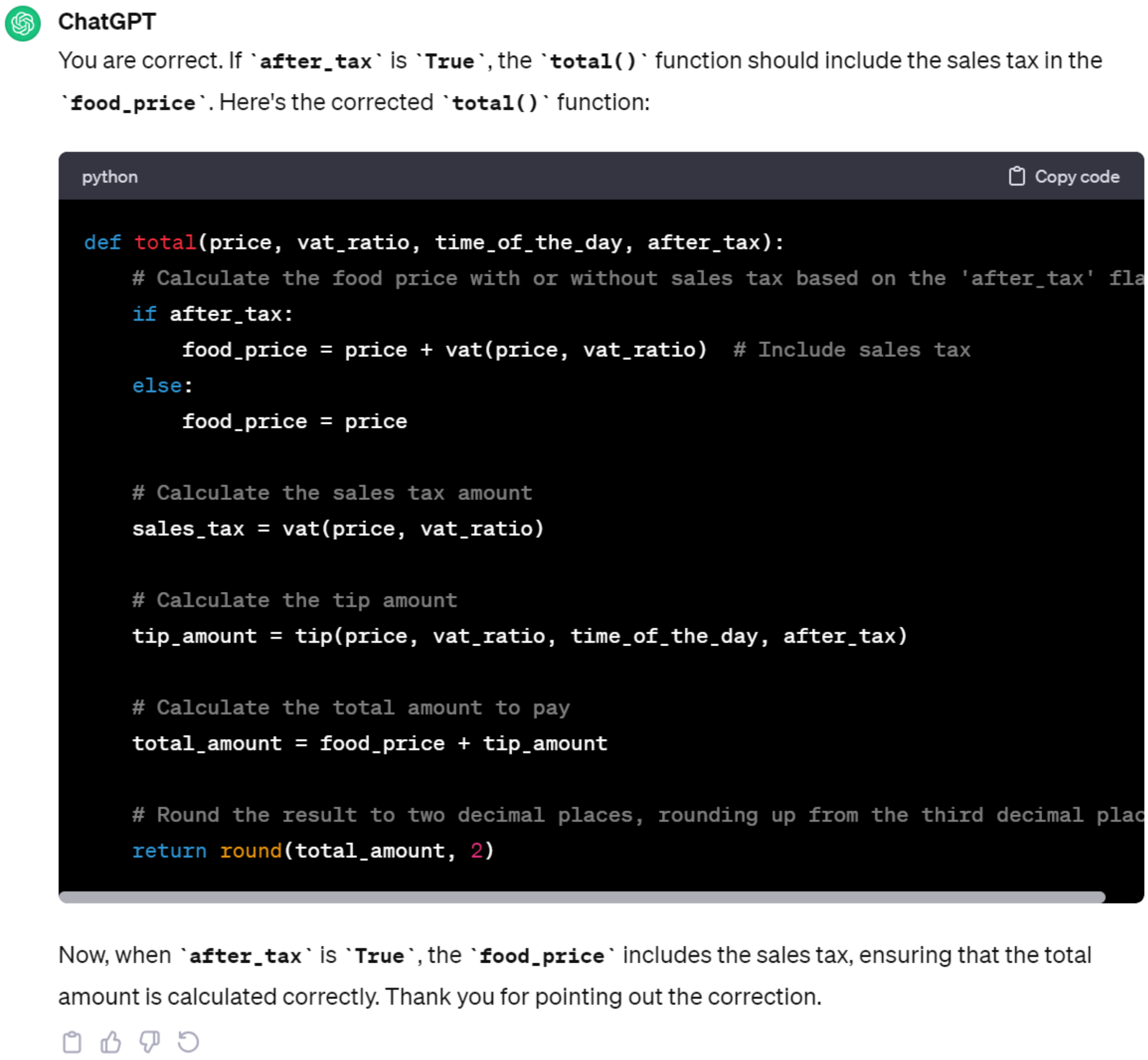


You

In the total function, if after_tax is True you should include sales_tax in food_price

Lastly, I requested to include sales_tax in food_price .If after_tax is True, sales_tax should be included in food_price. But in this script, it does not work. Because as you can see, in the original script, the if condition and the following code are mismatched.

```
if after_tax:
        food_price = price
    else:
        food_price = price * (1 + vat_ratio / 100)
```



ChatGPT

You are correct. If `after_tax` is `True`, the `total()` function should include the sales tax in the `food_price`. Here's the corrected `total()` function:

```python
def total(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the food price with or without sales tax based on the 'after_tax' fla
    if after_tax:
        food_price = price + vat(price, vat_ratio)  # Include sales tax
    else:
        food_price = price

    # Calculate the sales tax amount
    sales_tax = vat(price, vat_ratio)

    # Calculate the tip amount
    tip_amount = tip(price, vat_ratio, time_of_the_day, after_tax)

    # Calculate the total amount to pay
    total_amount = food_price + tip_amount

    # Round the result to two decimal places, rounding up from the third decimal plac
    return round(total_amount, 2)
```

Now, when `after_tax` is `True`, the `food_price` includes the sales tax, ensuring that the total amount is calculated correctly. Thank you for pointing out the correction.

So, ChatGPT fixed the script matching the following code to the if condition correctly. Now when after_tax is True, sales_tax is c=included in food_price.

```
if after_tax:
     food_price = price + vat(price, vat_ratio)  # Include sales tax
 else:
     food_price = price
```

Script

```
import math


def vat(price, vat_ratio):
    # Calculate the sales tax amount
    sales_tax = price * vat_ratio
    # Round the result to two decimal places, rounding up from the third decimal place
    return round(sales_tax, 2)


def tip(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the tip percentage based on the time of day
    if 5 <= time_of_the_day < 16:
        tip_percentage = 0.15
    else:
        tip_percentage = 0.20

    # Calculate the tip amount
    if after_tax:
        # Calculate tip based on the food price including sales tax
        tip_amount = (price + vat(price, vat_ratio)) * tip_percentage
    else:
        # Calculate tip based on the food price without sales tax
        tip_amount = price * tip_percentage

    # Round the result to two decimal places, rounding up from the third decimal place
    return round(tip_amount, 2)

def total(price, vat_ratio, time_of_the_day, after_tax):
    # Calculate the food price with or without sales tax based on the 'after_tax' flag
    if after_tax:
        food_price = price + vat(price, vat_ratio)  # Include sales tax
    else:
        food_price = price

    # Calculate the sales tax amount
    sales_tax = vat(price, vat_ratio)

    # Calculate the tip amount
    tip_amount = tip(price, vat_ratio, time_of_the_day, after_tax)

    # Calculate the total amount to pay
    total_amount = food_price + sales_tax + tip_amount
    # Round the result to two decimal places, rounding up from the third decimal place

    return round(total_amount, 2)
```

Conclusion

This project provided an opportunity to explore methods for solving problems encountered while using ChatGPT.

Firstly, in addressing the issues, the entire script was divided into individual functions, solved, and then implemented in sequence. This approach significantly reduced the likelihood of contextual confusion and fostered focused problem-solving, leading to an expected improvement in problem-solving skills. Had the modifications of all three functions been requested simultaneously, it would have taken much longer to resolve the errors. This method can be seen as a kind of Divide and Conquer strategy.

Additionally, it served as an opportunity to indirectly recognize the technical limitations of ChatGPT. After inputting the information from Table1, errors appeared in all three generated functions, each of a different nature. The problem with the first function arose in the process of handling context—a calculation error due to treating the variable 'vat_ratio' as one implying a percentage rather than a ratio. It is presumed that the error occurred in the scripting process due to a misunderstanding of the variable name 'vat_ratio'. This issue was resolved by directing ChatGPT to interpret 'vat_ratio' as a ratio, not a percentage.

The errors in the second and third functions were unique, with both occurring in if statements. The error in the second function, 'tip', was caused by including an unrelated variable in the condition statement, which should have affected only one decision. Similarly, the error in the third function, 'total', occurred in the mapping process of conditions and decisions. A commonality in these issues was the handling of the Boolean type variable 'after_tax', suggesting that ChatGPT might be more vulnerable to errors with Boolean data than other types. The second error was resolved by requesting a separation in the conditions, and the third by indicating a mapping error in the conditions and decisions.

This process has been an opportunity to explore methodologies for resolving inevitable errors in using ChatGPT and to speculate on the types of errors that occur and their underlying technical flaws. It is anticipated that as ChatGPT continues to evolve, such issues will diminish.