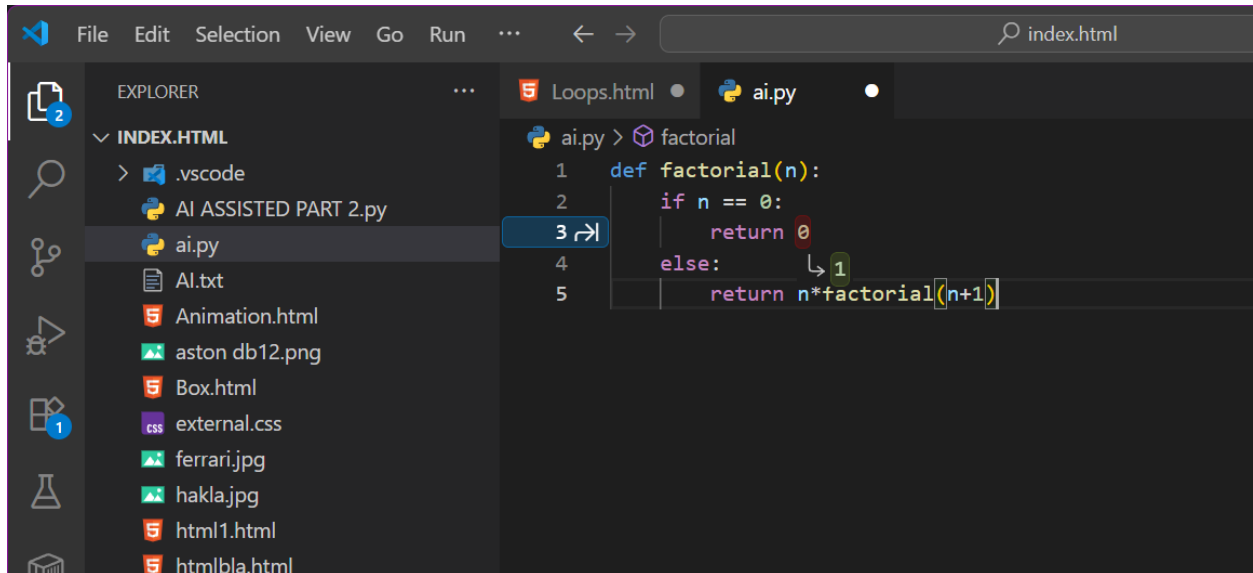
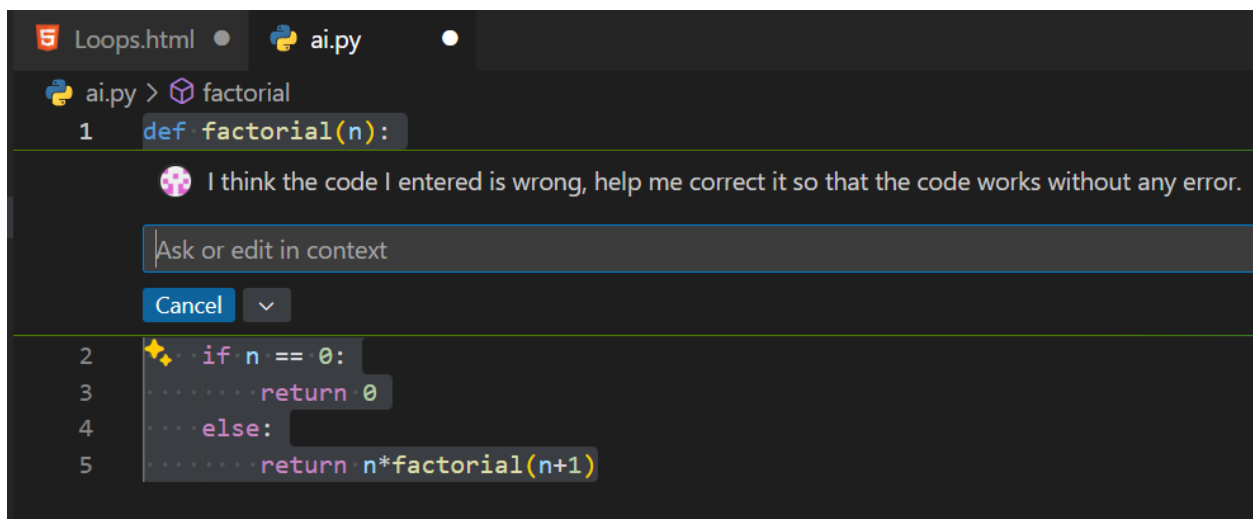


TASK #1

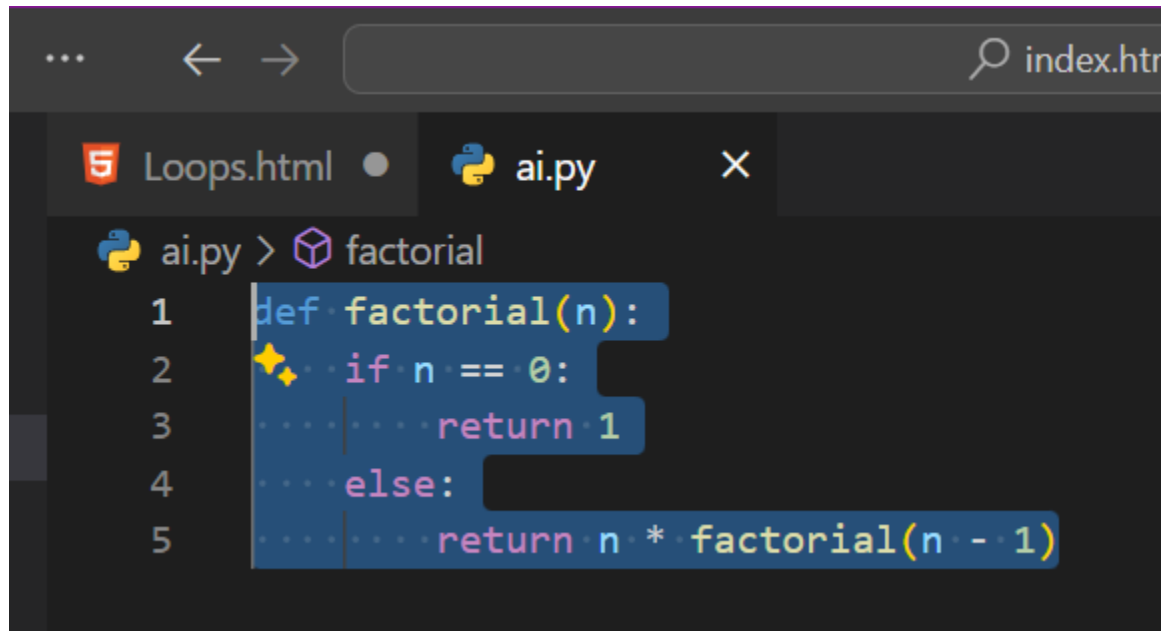
Input(BUGGY CODE):



Prompt given to correct the code:



Output (Corrected Code):

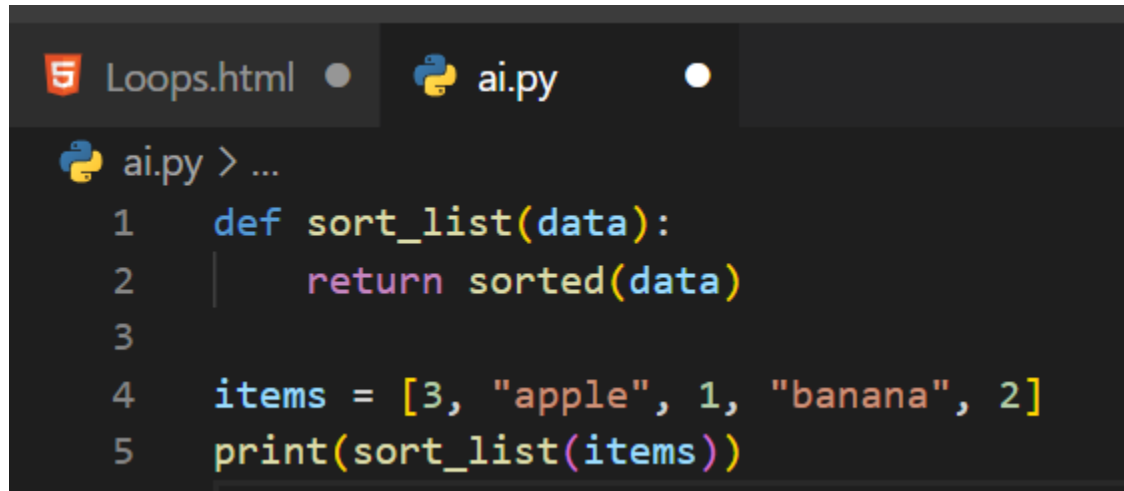


The image shows a code editor window with two tabs: 'Loops.html' and 'ai.py'. The 'ai.py' tab is active, displaying a Python function definition for 'factorial'. The code is as follows:

```
ai.py > factorial
1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n - 1)
```

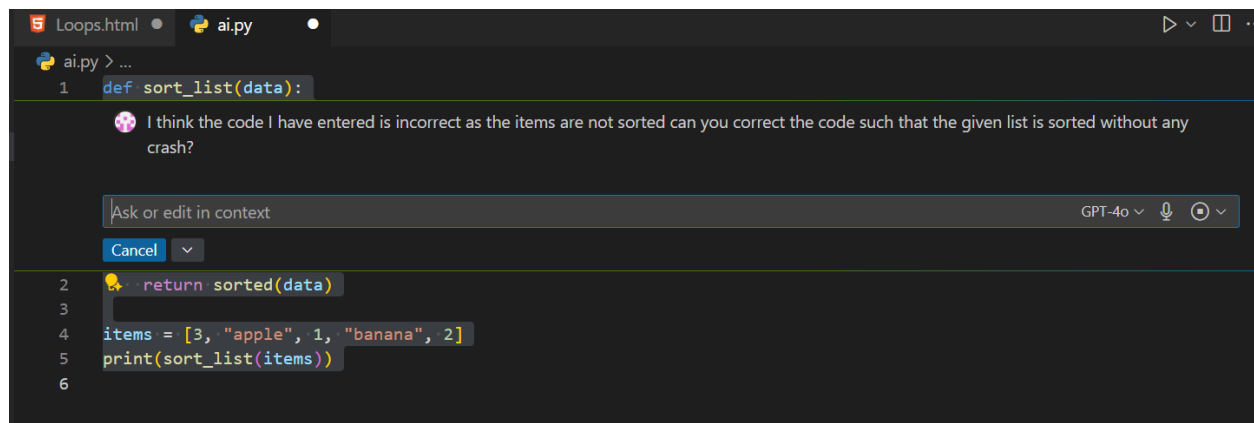
TASK #2

Input (Buggy Code):



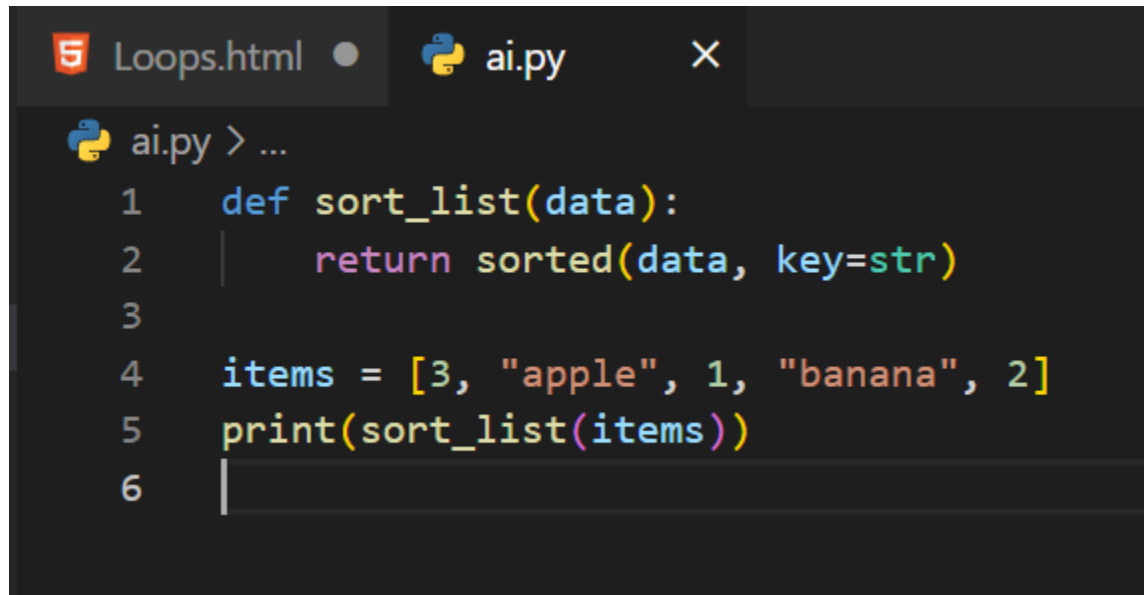
```
Loops.html • ai.py  
ai.py > ...  
1 def sort_list(data):  
2     return sorted(data)  
3  
4 items = [3, "apple", 1, "banana", 2]  
5 print(sort_list(items))
```

Prompt Given:



```
Loops.html • ai.py  
ai.py > ...  
1 def sort_list(data):  
    I think the code I have entered is incorrect as the items are not sorted can you correct the code such that the given list is sorted without any crash?  
  
    Ask or edit in context GPT-4o  
    Cancel  
2     return sorted(data)  
3  
4 items = [3, "apple", 1, "banana", 2]  
5 print(sort_list(items))  
6
```

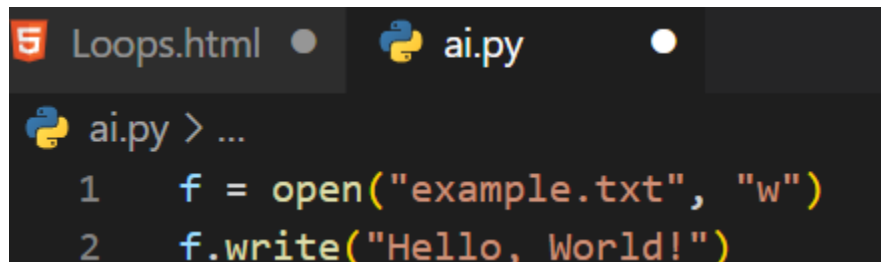
Output (Corrected Code):



```
Loops.html ● ai.py ×  
ai.py > ...  
1 def sort_list(data):  
2     return sorted(data, key=str)  
3  
4 items = [3, "apple", 1, "banana", 2]  
5 print(sort_list(items))  
6
```

Task #3

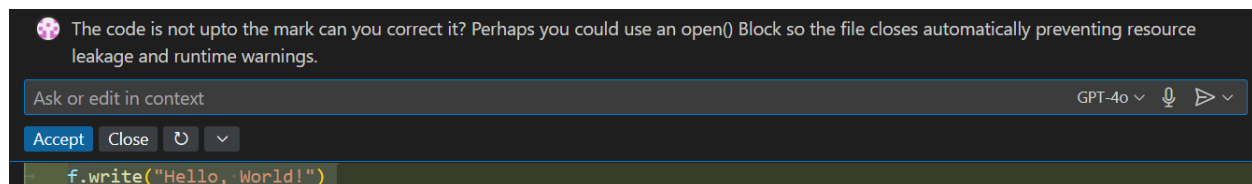
Input (Buggy Code):



The screenshot shows a code editor with two tabs: 'Loops.html' and 'ai.py'. The 'ai.py' tab is active, displaying the following Python code:

```
ai.py > ...  
1 f = open("example.txt", "w")  
2 f.write("Hello, World!")
```

Prompt Given :



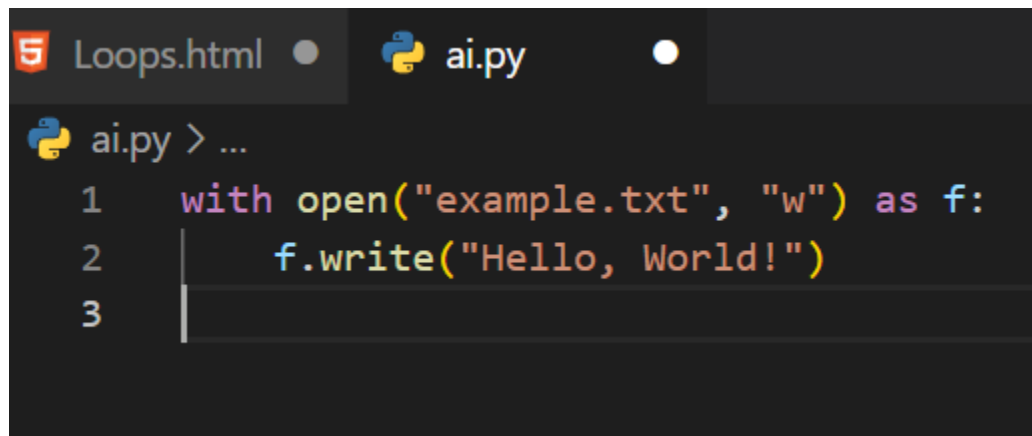
The screenshot shows a chat interface with a message from a user and a response from an AI model. The user's message is:

The code is not upto the mark can you correct it? Perhaps you could use an open() Block so the file closes automatically preventing resource leakage and runtime warnings.

The AI model's response is:

```
f.write("Hello, World!")
```

Output (Corrected Code):



The screenshot shows the same code editor as before, but now the 'ai.py' tab contains the corrected Python code:

```
ai.py > ...  
1 with open("example.txt", "w") as f:  
2     f.write("Hello, World!")  
3
```

Task #4

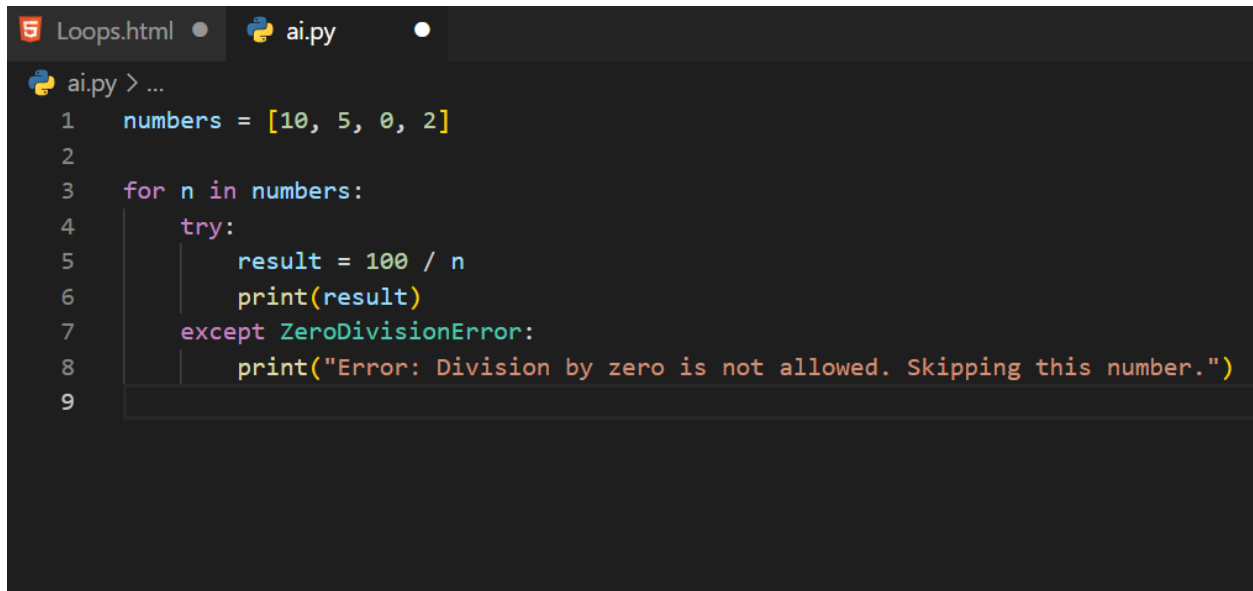
Input (Buggy Code):

```
ai.py > ...  
1  numbers = [10, 5, 0, 2]  
2  
3  for n in numbers:  
4      result = 100 / n  
5      print(result)  
6
```

Prompt Given:

```
ai.py > ...  
1  numbers = [10, 5, 0, 2]  
  
In order to improve the current state of the code, implement error handling using try-except and continue the execution safely. The main motive to implement a try except block is to prevent crashes and printing a meaningful error message for the user. GPT-4o v  
user:  
  
2    
3  for n in numbers:  
4      result = 100 / n  
5      print(result)  
6
```

Output:



```
Loops.html ● ai.py ●  
ai.py > ...  
1 numbers = [10, 5, 0, 2]  
2  
3 for n in numbers:  
4     try:  
5         result = 100 / n  
6         print(result)  
7     except ZeroDivisionError:  
8         print("Error: Division by zero is not allowed. Skipping this number.")  
9
```



Task #5

Input (Buggy Code):

```
Loops.html • ai.py 1 •
ai.py > ...
1 class Person:
2     def __init__(self, name, age):
3         self.fullname = fullname
4         self.agee = age
5
6     def greet(self):
7         return f"Hello, my name is {self.name} and I am {self.age} years old."
8
```

Prompt given:

The code seems buggy, analyze and correct the constructor and attribute usage, in a way you should identifies mismatched paramater or missing self references and rewrite the class with accurate initialisation.

GPT-4o v  

Output:

```
ai.py > ...
1 class Person:
2     def __init__(self, name, age):
3         self.name = name
4         self.age = age
5
6     def greet(self):
7         return f"Hello, my name is {self.name} and I am {self.age} years old."
8
```