

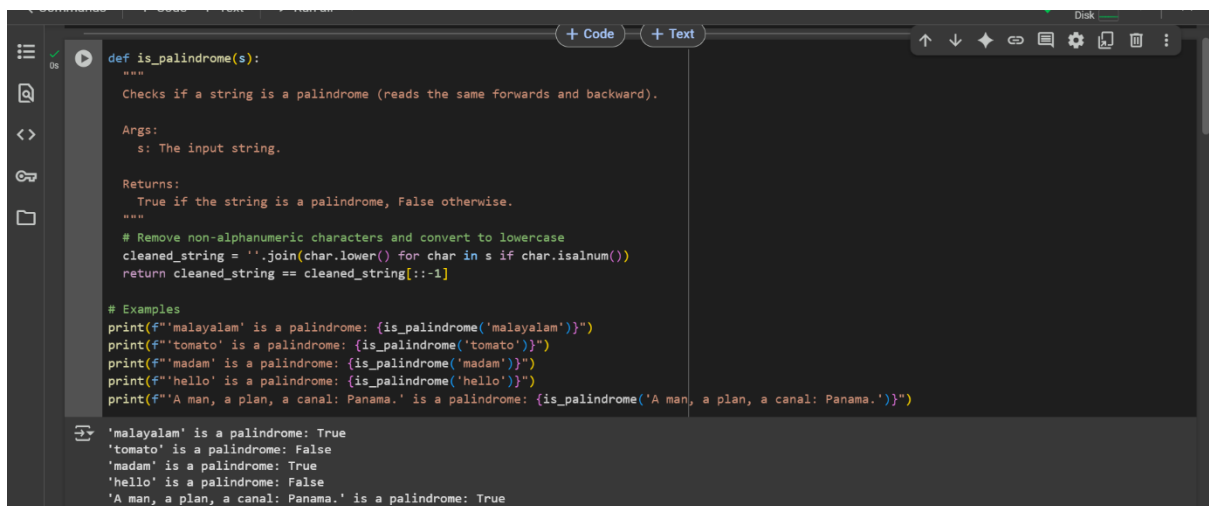
LAB TEST: 1

Name: Bhanuchander

Roll No: 2403A52118

1; Write a comment prompt with 2-3 examples asking copilot (gemini; to generate a python function that checks whether a string is a palindrome

Code and output;



```
def is_palindrome(s):
    """
    Checks if a string is a palindrome (reads the same forwards and backward).

    Args:
        s: The input string.

    Returns:
        True if the string is a palindrome, False otherwise.
    """
    # Remove non-alphanumeric characters and convert to lowercase
    cleaned_string = ''.join(char.lower() for char in s if char.isalnum())
    return cleaned_string == cleaned_string[::-1]

# Examples
print(f"'malayalam' is a palindrome: {is_palindrome('malayalam')}")
print(f"'tomato' is a palindrome: {is_palindrome('tomato')}")
print(f"'madam' is a palindrome: {is_palindrome('madam')}")
print(f"'hello' is a palindrome: {is_palindrome('hello')}")
print(f"'A man, a plan, a canal: Panama.' is a palindrome: {is_palindrome('A man, a plan, a canal: Panama.')}")
```

'malayalam' is a palindrome: True
'tomato' is a palindrome: False
'madam' is a palindrome: True
'hello' is a palindrome: False
'A man, a plan, a canal: Panama.' is a palindrome: True

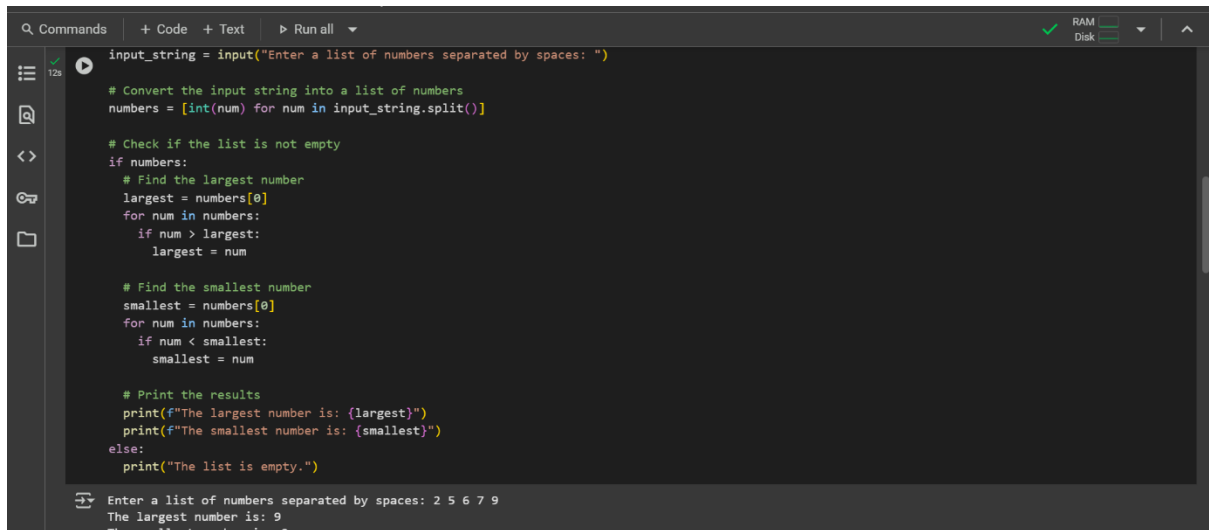
Explanation;

- 1. Clean up the input:** It takes the word or sentence and removes anything that's not a letter or a number. It also makes all the letters lowercase. So, "A man, a plan, a canal: Panama." becomes "amanaplanacanalpanama".
- 2. Compare forward and backward:** It then checks if this cleaned-up version of the word or sentence is exactly the same as the cleaned-up version read backward. If they match, it's a palindrome!

The examples show how this works with different words and phrases.

2; ask google copilot (gemini to generate a python program that finds the largest and smallest number in a user provided list then, ask gemini to explain how the code works

Code and output;



```
input_string = input("Enter a list of numbers separated by spaces: ")

# Convert the input string into a list of numbers
numbers = [int(num) for num in input_string.split()]

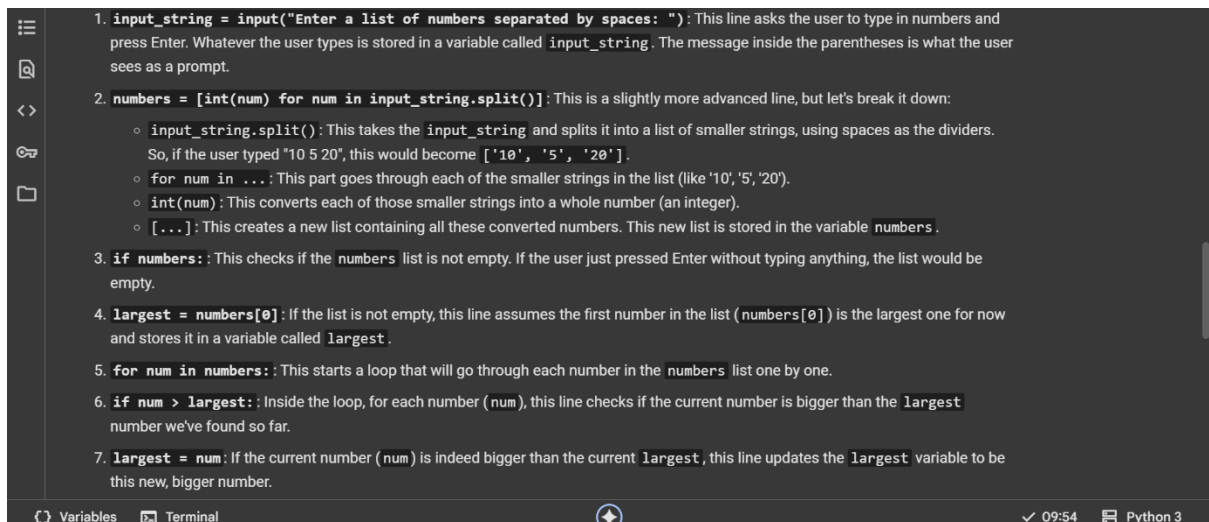
# Check if the list is not empty
if numbers:
    # Find the largest number
    largest = numbers[0]
    for num in numbers:
        if num > largest:
            largest = num

    # Find the smallest number
    smallest = numbers[0]
    for num in numbers:
        if num < smallest:
            smallest = num

    # Print the results
    print(f"The largest number is: {largest}")
    print(f"The smallest number is: {smallest}")
else:
    print("The list is empty.")
```

Enter a list of numbers separated by spaces: 2 5 6 7 9
The largest number is: 9
The smallest number is: 2

Explanation;



```
1. input_string = input("Enter a list of numbers separated by spaces: "): This line asks the user to type in numbers and
   press Enter. Whatever the user types is stored in a variable called input_string. The message inside the parentheses is what the user
   sees as a prompt.

2. numbers = [int(num) for num in input_string.split()]: This is a slightly more advanced line, but let's break it down:
   - input_string.split(): This takes the input_string and splits it into a list of smaller strings, using spaces as the dividers.
     So, if the user typed "10 5 20", this would become ['10', '5', '20'].
   - for num in ...: This part goes through each of the smaller strings in the list (like '10', '5', '20').
   - int(num): This converts each of those smaller strings into a whole number (an integer).
   - [...]: This creates a new list containing all these converted numbers. This new list is stored in the variable numbers.

3. if numbers: This checks if the numbers list is not empty. If the user just pressed Enter without typing anything, the list would be
   empty.

4. largest = numbers[0]: If the list is not empty, this line assumes the first number in the list (numbers[0]) is the largest one for now
   and stores it in a variable called largest.

5. for num in numbers: This starts a loop that will go through each number in the numbers list one by one.

6. if num > largest: Inside the loop, for each number (num), this line checks if the current number is bigger than the largest
   number we've found so far.

7. largest = num: If the current number (num) is indeed bigger than the current largest, this line updates the largest variable to be
   this new, bigger number.
```

Variables Terminal 09:54 Python 3



```
8. smallest = numbers[0]: Similar to finding the largest, this line assumes the first number is the smallest for now and stores it in a variable called smallest.  
9. for num in numbers:: This starts another loop, just like before, to go through each number in the list.  
10. if num < smallest:: Inside this loop, it checks if the current number (num) is smaller than the smallest number we've found so far.  
11. smallest = num: If the current number (num) is smaller than the current smallest, this line updates the smallest variable to be this new, smaller number.  
12. print(f"The largest number is: {largest}"): After checking all the numbers, this line prints out the final value of the largest number found. The f"" is a way to easily include the value of the largest variable directly in the text.  
13. print(f"The smallest number is: {smallest}"): This line does the same for the smallest number.  
14. else:: This part of the code runs if the if numbers: check in step 3 was False, meaning the list was empty.  
15. print("The list is empty."): This line simply prints a message telling the user that the list they entered was empty.
```