

Assignment 2

Task-1:

simple [python code to read the list of numbers and calculate the mean, minimum and maximum values using functions

The screenshot shows a VS Code editor with a Python file named 'ass 2.1.py'. The code defines three functions: `calculate_mean`, `calculate_min`, and `calculate_max`. It then reads input from the user, splits it into a list of floats, and calls these functions to calculate the mean, minimum, and maximum of the list. The output is printed to the console.

```
1 # simple python code to read a list of numbers and calculate mean, minimum and maximum values using functions
2
3 def calculate_mean(numbers):
4     return sum(numbers) / len(numbers) if numbers else 0
5
6 def calculate_min(numbers):
7     return min(numbers) if numbers else None
8
9 def calculate_max(numbers):
10    return max(numbers) if numbers else None
11
12 # Read numbers from user
13 numbers = input("Enter numbers separated by spaces: ").split()
14 numbers = [float(num) for num in numbers]
15
16 mean = calculate_mean(numbers)
17 minimum = calculate_min(numbers)
18 maximum = calculate_max(numbers)
19
20 print(f"Mean: {mean}")
21 print(f"Minimum: {minimum}")
22 print(f"Maximum: {maximum}")
```

The terminal output shows the execution of the script:

```
PS C:\Aiassisted coding> & 'c:\Users\Poojasree\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\Poojasree\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '59549' '--' 'C:\Aiassisted coding\ass 2.1.py'
Enter numbers separated by spaces: 1 3 2 4 5 7 8 5 3 2
Mean: 4.0
Minimum: 1.0
Maximum: 8.0
PS C:\Aiassisted coding>
```

The right sidebar shows a reference to the code and an explanation of the code:

Used 1 reference

Here's an explanation of the code:

- calculate_mean(numbers):**
 - Returns the average of the list by dividing the sum by the count.
 - If the list is empty, returns 0.
- calculate_min(numbers):**
 - Returns the smallest value in the list.
 - If the list is empty, returns None.
- calculate_max(numbers):**
 - Returns the largest value in the list.
 - If the list is empty, returns None.
- Reading Input:**
 - Prompts the user to enter numbers separated by spaces.
 - Splits the input string into a list and converts each item to a float.
- Calculations:**
 - Calls the three functions to get mean, minimum, and maximum values.
- Output:**
 - Prints the calculated mean, minimum, and maximum values.

Task-2:

A simple python code to check whether a given number is armstrong or not

```
1 # A simple python code to check whether a given number is armstrong or not
2
3 def is_armstrong_number(num):
4     # Convert the number to string to easily iterate over digits
5     digits = str(num)
6     power = len(digits)
7     return sum(int(digit) ** power for digit in digits) == num
8
9 # Read a number from user
10 number = int(input("Enter a number: "))
11
12 if is_armstrong_number(number):
13     print(f"{number} is an Armstrong number.")
14 else:
15     print(f"{number} is not an Armstrong number.")
16
```

if is_armstrong_number(number):
print(f"{number} is an Armstrong number.")
else:
print(f"{number} is not an Armstrong number.") explain the code

ass2.2.py:3-15

> Used 1 reference

Here's an explanation of the code:

- **is_armstrong_number(num):**
 - Converts the number to a string to access each digit.
 - Calculates the number of digits (`len` power).
 - Sums each digit raised to the power of the number of digits.
 - Returns `True` if the sum equals the original number (Armstrong number), otherwise `False`.
- **User Input:**
 - Reads an integer from the user.
- **Check and Output:**
 - Calls the function to check if the number is Armstrong.
 - Prints whether the number is an Armstrong number or not.

Armstrong number:
A number is Armstrong if the sum of its digits each raised to the power of the number of digits equals the number itself (e.g., $153 \rightarrow 1^3 + 5^3 + 3^3 = 153$).

Ass2.2.py X

Add context (#), extensions (@), commands (/)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python Debug Console + - [] [X] [] [X]

```
ng\ass2.2.py'
Enter a number: 153
153 is an Armstrong number.
PS C:\Aiassisted coding> ^C
PS C:\Aiassisted coding>
PS C:\Aiassisted coding> c:: cd 'c:\Aiassisted coding'; & 'c:\Users\Poojasree\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\Poojasree\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '59788' '--' 'C:\Aiassisted coding\ass2.2.py'
Enter a number: 124
124 is not an Armstrong number.
PS C:\Aiassisted coding>
```

Task-3:

simple code in python to check whether given number is a prime or not

The image shows a VS Code editor with a Python file named `ass2.3.py`. The code defines a function `is_prime` and uses it to check if a user-input number is prime. The right-hand sidebar shows an AI-generated explanation of the code.

```
1 #simple code in python to check whether given number is a prime or not
2
3 def is_prime(num):
4     if num <= 1:
5         return False
6     for i in range(2, int(num**0.5) + 1):
7         if num % i == 0:
8             return False
9     return True
10
11 # Read a number from user
12 number = int(input("Enter a number: "))
13
14 if is_prime(number):
15     print(f"{number} is a prime number.")
16 else:
17     print(f"{number} is not a prime number.")
18
```

if is_prime(number):
print(f"{number} is a prime number.")
else:
print(f"{number} is not a prime number.") explain the code

> Used 1 reference

Here's an explanation of the code:

- **is_prime(num):**
 - Checks if the input number is less than or equal to 1. If so, returns `False` (not prime).
 - Loops from 2 up to the square root of the number (`int(num**0.5) + 1`).
 - If the number is divisible by any value in this range, returns `False` (not prime).
 - If no divisors are found, returns `True` (the number is prime).
- **User Input:**
 - Reads an integer from the user.
- **Prime Check and Output:**
 - Calls `is_prime` to check if the number is prime.
 - Prints whether the number is prime or not.

Summary:
This code checks if a given number is prime by testing divisibility up to its square root and prints the result.

PS C:\Aiassisted coding> & 'c:\Users\Poojasree\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\Poojasree\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '59927' '--' 'C:\Aiassisted coding\ass2.3.py'
Enter a number: 3
3 is a prime number.
PS C:\Aiassisted coding> ^C
PS C:\Aiassisted coding>
PS C:\Aiassisted coding> c:: cd 'c:\Aiassisted coding'; & 'c:\Users\Poojasree\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\Poojasree\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '59947' '--' 'C:\Aiassisted coding\ass2.3.py'
Enter a number: 4
4 is not a prime number.
PS C:\Aiassisted coding>

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS Python Debug Console

Add Context... ass2.3.py X

Add context (#), extensions (@), commands (/)

Ask GPT-4.1

Task-4:

Simple code in python for sum of 1st n natural numbers

The image shows a VS Code editor with a Python file named `ass2.4.py`. The code defines a function `sum_of_natural_numbers(n)` that calculates the sum of the first `n` natural numbers using the formula $n * (n + 1) // 2$. It then prompts the user to enter a number and prints the result.

```
1 # simple code for python for sum of first n natural numbers
2 def sum_of_natural_numbers(n):
3     return n * (n + 1) // 2
4
5 # Read a number from user
6 number = int(input("Enter a number: "))
7 print(f"Sum of first {number} natural numbers is: {sum_of_natural_numbers(number)}")
8
```

The terminal output shows the command to run the script, the user input `10`, and the output `Sum of first 10 natural numbers is: 55`.

Below the code, an AI-generated explanation is provided:

Read a number from user

```
number = int(input("Enter a number: "))
print(f"Sum of first {number} natural numbers is: {sum_of_natural_numbers(number)}")
explain the code
```

Used 1 reference

Here's an explanation of the code:

- **sum_of_natural_numbers(n):**
 - Calculates the sum of the first `n` natural numbers using the formula `n * (n + 1) // 2`.
 - This formula quickly gives the sum without needing a loop.
- **User Input:**
 - Reads an integer from the user.
- **Output:**
 - Prints the sum of the first `number` natural numbers using the function.

Summary:

The code efficiently computes and displays the sum of the first `n` natural numbers entered by the user.

Task-5:

simple code in python to sum of odd numbers and even numbers in the given tuple

```
ass.2.5.py > ...
1 # simple code in python to sum of odd numbers and even numbers in the given tuple
2 def sum_of_odd_even(numbers):
3     odd_sum = sum(num for num in numbers if num % 2 != 0)
4     even_sum = sum(num for num in numbers if num % 2 == 0)
5     return odd_sum, even_sum
6
7 # Read a tuple from user
8 user_input = input("Enter numbers separated by spaces: ")
9 numbers = tuple(int(num) for num in user_input.split())
10
11 odd_sum, even_sum = sum_of_odd_even(numbers)
12 print(f"Sum of odd numbers: {odd_sum}")
13 print(f"Sum of even numbers: {even_sum}")
14
```

simple code in python to sum of odd numbers and even numbers in the given tuple explain this code

ass.2.5.py:1

> Used 1 reference

Here's an explanation of the code:

1. Function Definition

```
def sum_of_odd_even(numbers):
    odd_sum = sum(num for num in numbers if num % 2 != 0)
    even_sum = sum(num for num in numbers if num % 2 == 0)
    return odd_sum, even_sum
```

- This function takes a tuple of numbers.
- It calculates the sum of odd numbers (`num % 2 != 0`) and even numbers (`num % 2 == 0`) using generator expressions.
- Returns both sums.

2. User Input

```
user_input = input("Enter numbers separated by spaces: ")
numbers = tuple(int(num) for num in user_input.split())
```

- Prompts the user to enter numbers separated by spaces.

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Python Debug Console + - [] [x] [] [x]

```
PS C:\Aiassisted coding> & 'c:\Users\Poojasree\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\Poojasree\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '60291' '--' 'C:\Aiassisted coding\ass.2.5.py'
Enter numbers separated by spaces: 1 3 2 4 6 7 8 12 15 23
Sum of odd numbers: 49
Sum of even numbers: 32
PS C:\Aiassisted coding>
```