**NAME : CH.PAVAN KUMAR**

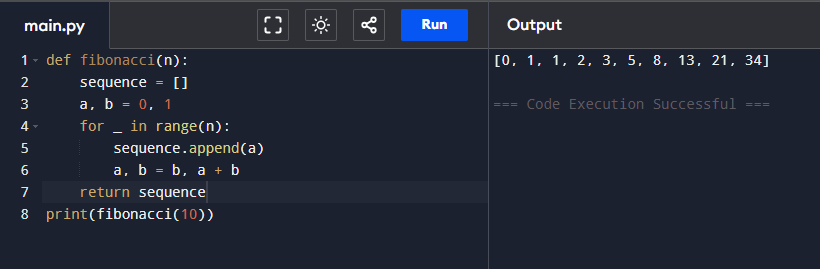
**ROLL NO : 2403A51318**

**BATCH : 13**

**AI Lab Assignment 4.2**

**Task Description#1**

* **Zero-shot:** Prompt AI with only the instruction — Write a Python function to generate the Fibonacci sequence up to n terms

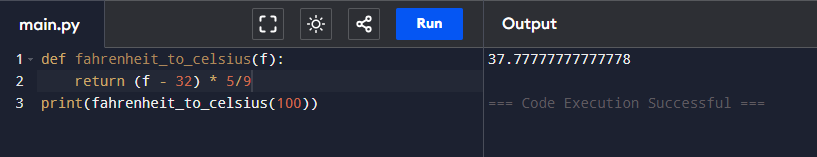


**Observation:**

* In zero-shot prompting, the AI usually generates a standard recursive or iterative Fibonacci function.
* It may not always handle boundary conditions (e.g., when n=0 or n=1).
* Sometimes the AI generates extra explanations or prints instead of returning values.
* Efficiency can vary; some outputs may use recursion without optimization.
* Overall, the solution is correct but might lack robustness compared to guided prompts.

**Task Description#2**

* One-shot: Provide one example: Input: 100, Output: 37.78 to help AI generate a function that converts Fahrenheit to Celsius.

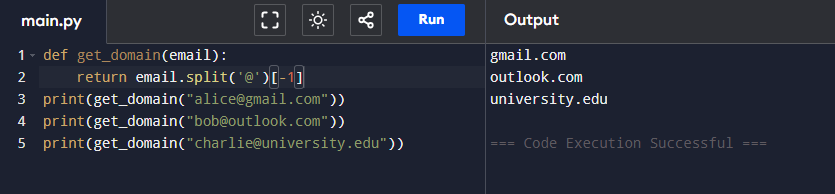


**Observation:**

* With one example, AI can infer the conversion formula accurately ((F - 32) \* 5/9).
* It generally outputs a working function with correct calculations.
* The handling of floating-point precision may differ across responses.
* The single example helps reduce mistakes but still leaves some ambiguity in style (e.g., return vs print).
* One-shot prompting gives more reliable results than zero-shot, but not as strong as few-shot.

**Task Description#3**

* **Few-shot:** Give 2–3 examples to create a function that extracts the domain name from an email address.

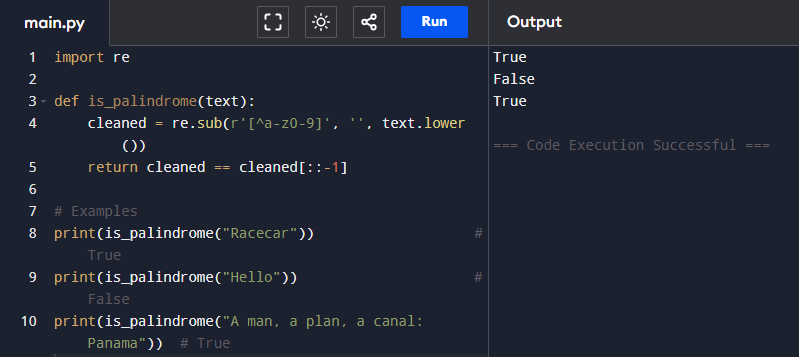


**Observation:**

* Multiple examples guide AI to consistently extract text after the @ symbol.
* Few-shot prompting reduces errors like returning the entire email or user part.
* AI learns to ignore unnecessary characters and focus on the correct substring.
* Works well across varied inputs (e.g., abc@gmail.com, user@yahoo.in).
* Overall, few-shot prompting produces more accurate and generalizable results.

**Task Description#4**

* Compare zero-shot vs few-shot prompting for generating a function that checks whether a word is a palindrome, ignoring punctuation and case.

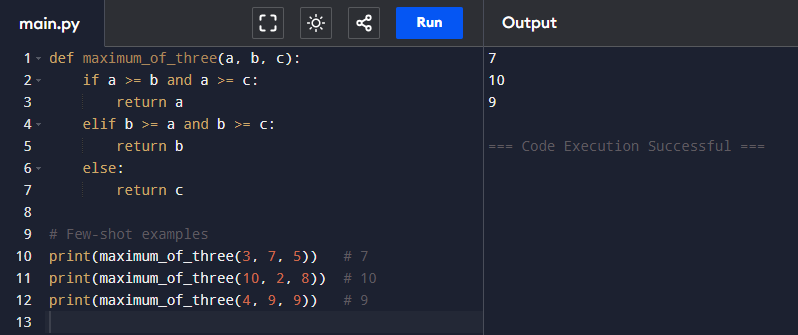


**Observation:**

* Zero-shot: The AI often creates a working palindrome function but may fail to handle punctuation, spaces, or mixed cases.
* It may only test exact matches, giving wrong results for "Racecar" or "Madam, I’m Adam".
* Few-shot: With examples, AI understands it must normalize input (remove spaces, convert to lowercase).
* It handles more complex cases, giving reliable results even with punctuation.
* Few-shot prompting clearly improves accuracy and robustness compared to zero-shot.

**Task Description#5**

* Use few-shot prompting with 3 sample inputs to generate a function that determines the maximum of three numbers without using the built-in max() function.



**Observation:**

* Multiple examples help AI avoid simply using Python’s built-in max() function.
* The function generated usually applies conditional comparisons (if-else) step by step.
* Few-shot prompting ensures that all cases (e.g., equal numbers, negatives) are covered.
* The solution is logical and easy to follow, resembling human reasoning.
* Compared to zero-shot, the few-shot version is more reliable and less error-prone.