Nasrulla M U  
IT24101774

Exercise 2

1. Where does your favourite model rank currently?

Answer : 7th

1. Try out at least two models that you haven't tried before (you will have to do a seperate google search to find these products)
2. Run two prompts of your choice
3. Submit the model, the prompt and the responses you have got.

#### **Model 1: DeepSeek-R1**

* **Prompt:** Think step by step to solve the following problem: A farmer has 120 meters of fencing and wants to create a rectangular enclosure for his chickens. He will use an existing wall for one side of the enclosure. What dimensions should he use to maximize the area of the enclosure? Show your work.
* **Response:**
  + **Step 1:** Define the variables. Let the two sides perpendicular to the wall be x and the side parallel to the wall be y.
  + **Step 2:** Formulate the equations. The perimeter is given by the fencing used, so 2x + y = 120. The area is given by A = x \* y.
  + **Step 3:** Express Area in terms of one variable. From the perimeter equation, y = 120 - 2x. Substitute this into the area equation: A(x) = x \* (120 - 2x) = 120x - 2x^2.
  + **Step 4:** Find the maximum area using calculus. To find the maximum, take the derivative of the area function and set it to zero: A'(x) = 120 - 4x. 120 - 4x = 0, which means 4x = 120, so x = 30.
  + **Step 5:** Calculate the other dimension. Substitute x = 30 back into the perimeter equation: 2(30) + y = 120, which means 60 + y = 120, so y = 60.
  + **Step 6:** State the final answer. The dimensions that maximize the area are **30 meters by 60 meters**. The maximum area would be 1800 square meters.

#### **Model 2: Qwen3 Coder**

* **Prompt:** Write a Python function that takes a list of integers as input and returns a new list containing only the unique elements from the original list, while maintaining their original order. Do not use a set. Provide a simple example of how to use the function.

**Response:**Python  
def get\_unique\_elements\_ordered(input\_list):

"""

Returns a list of unique elements from the input list,

maintaining their original order.

Args:

input\_list (list): A list of integers.

Returns:

list: A new list with only the unique elements.

"""

unique\_list = []

seen = {} # Use a dictionary as a hash map for efficient lookups

for item in input\_list:

if item not in seen:

seen[item] = True

unique\_list.append(item)

return unique\_list

# Simple example of how to use the function

my\_list = [1, 5, 2, 8, 5, 1, 9, 2]

unique\_elements = get\_unique\_elements\_ordered(my\_list)

print(f"Original list: {my\_list}")

print(f"Unique elements (in order): {unique\_elements}")

* **Output:** Original list: [1, 5, 2, 8, 5, 1, 9, 2] Unique elements (in order): [1, 5, 2, 8, 9]