This problem involves implementing an epsilon-greedy K-arm bandit algorithm for a recommendation system, specifically to recommend fruits to a user.

The recommendation system aims to recommend fruits to a user, learn their preferences based on feedback, and improve future recommendations.

Implementation steps:

* Define a list of items to be recommended.
* Set an exploration parameter, which determines the probability of exploring (trying random fruits).
* Initialize arrays to store the estimated value of each item (fruit) and the count of how often each item has been recommended.
* Implement a method to select an action (recommend a fruit) based on the epsilon-greedy strategy:
  + With a probability of epsilon, select a random fruit (exploration).
  + With a probability of 1 - epsilon, select the fruit with the highest estimated reward (exploitation).
* Create a UserSimulator class to simulate user responses to the recommendations. The simulator has a predefined probability of liking each fruit.
* The simulator will provide feedback (like or dislike) based on these probabilities.
* After receiving user feedback (reward), update the estimated value of the recommended fruit using incremental mean:
  + Increase the count of how many times the fruit has been recommended.
  + Update the estimated value.
* Run the recommendation loop multiple times:
  + Recommend a fruit based on the current estimated values and epsilon-greedy strategy.
  + Simulate user feedback using the UserSimulator.
  + Update the bandit’s estimated values based on the feedback.
* After the loop, analyze the estimated values for each fruit to determine which fruit is most likely preferred by the user.
* Also, check how many times each fruit was recommended to show how exploration and exploitation were balanced.