TURING

Technical Writing Assessment

READ THE INSTRUCTIONS IN THESE SLIDES CAREFULLY. TAKE YOUR TIME.
YOU WILL ONLY GET 1 ATTEMPT ONCE YOU BEGIN THE ASSESSMENT

Overview

We are seeking talented individuals capable of using programming languages to train Large Language Models (LLMs) to solve problems. The ideal candidates should be able to:

- Clearly explain their strategies for problem-solving.
- Design practical solutions in code.
- Develop test cases to validate their solutions.
- Debug and refine their solutions for improvement.

Assessment Objective

- Assess your **English writing proficiency** to ensure you can produce high-quality rationales. This involves evaluating your ability to write clearly, effectively, and grammatically correct texts.
- Examine your capability to provide a coherent and comprehensive analysis of various model responses. This includes articulating the advantages and disadvantages of each response, **identifying objective differences**, and maintaining appropriate technical terminology throughout the discussion.
- Determine your **attention to detail** - a crucial aspect when comparing model responses. You must be adept at noticing even the most subtle differences to accurately rank responses from best to worst.

Instructions

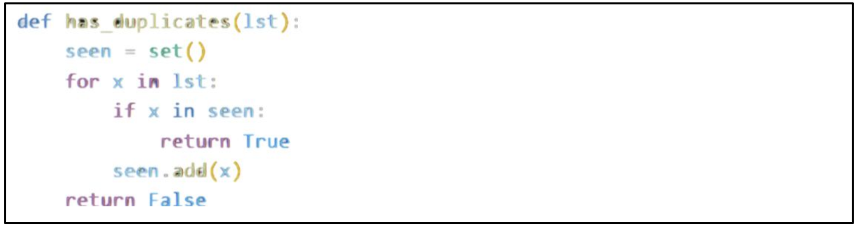
READ THESE INSTRUCTIONS CAREFULLY. YOU WILL ONLY GET 1 ATTEMPT.

- The total time for the assessment is **30 minutes.**
- You will be provided a short conversation based on interactions between a user and an AI model assistant.
- At the end of the conversation, there will be two potential responses given by the AI model to answer the user's last message/request.
- Based on the conversation, you are required to
  - Read the full interaction between the User and AI model assistant to understand the context.
  - Evaluate and compare the two potential responses by the AI model assistant, and write analyses considering:
    - **Correctness**: Consider the factuality, logical flow, contextual relevance, and consistency.
    - **Informativeness**: Is the information provided relevant to the user's request? Is the information complete?
    - **Which one would you choose as the best possible model response? Explain why.**
- 🚩DO NOT use your own IDE or any online debugger tools.
- 🚩DO NOT use ChatGPT or any other AI tool either to compare the responses or to improve your final response.

Good Example(1/2)

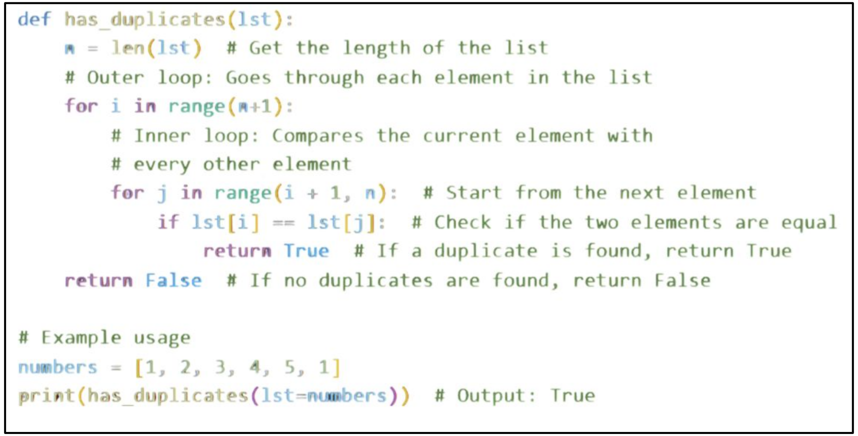
User: Give me a fast method in python to check if a list contains duplicated items.

Model response A:



```python
def has_duplicates(lst):
    seen = set()
    for x in lst:
        if x in seen:
            return True
        seen.add(x)
    return False
```

Model response B:



```python
def has_duplicates(lst):
    n = len(lst)  # Get the length of the list
    # Outer loop: Goes through each element in the list
    for i in range(n+1):
        # Inner loop: Compares the current element with
        # every other element
        for j in range(i + 1, n):  # Start from the next element
            if lst[i] == lst[j]:  # Check if the two elements are equal
                return True  # If a duplicate is found, return True
    return False  # If no duplicates are found, return False

# Example usage
numbers = [1, 2, 3, 4, 5, 1]
print(has_duplicates(lst=numbers))  # Output: True
```

Solution:

Correctness: Consider the factuality, logical flow, contextual relevance, and consistency.

Response A's approach provides code that has no major issues in satisfying the user's ask for python code to check if a list contains duplicated items. It is highly efficient due to leveraging Python's set, which only holds unique elements. This allows for constant-time membership checks (O(1)), ensuring quick operations for most dataset sizes.

Response B's approach provides code that has minor issues in satisfying the user's ask for python code to check if a list contains duplicated items. It is less efficient for larger lists, as it relies on comparing each element to every other, yielding a time complexity of O(n^2). Additionally, there's a minor error in using "for i in range(n+1)" instead of the correct "for i in range(n)".

Informativeness: Is the information provided relevant to the user's request? Is the information complete?

Response A's approach is not informative. It has a lack of comments, so a practical usage example might limit its immediate clarity for some users.

Response B's approach is very informative. Overall extensive commenting enhances its readability, aiding beginner understanding.

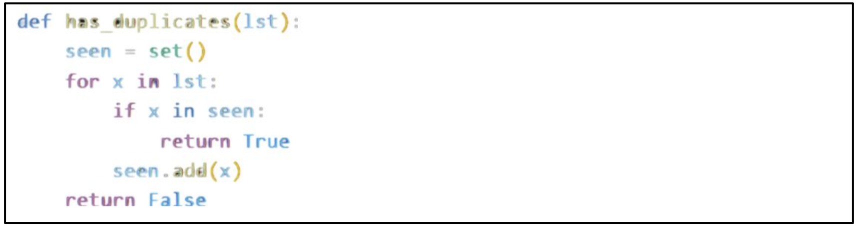
Which one would you choose as the best possible model response? Explain why.

Response A is a better response as it has no issues in correctness and more efficient. Response A is the preferred way to check for duplicates in most cases. Response B, while simpler and easier to understand, can be impractically slow for lists with many elements. The user is looking for a fast solution, so model response A is the best of the two.

Good Example(2/2)

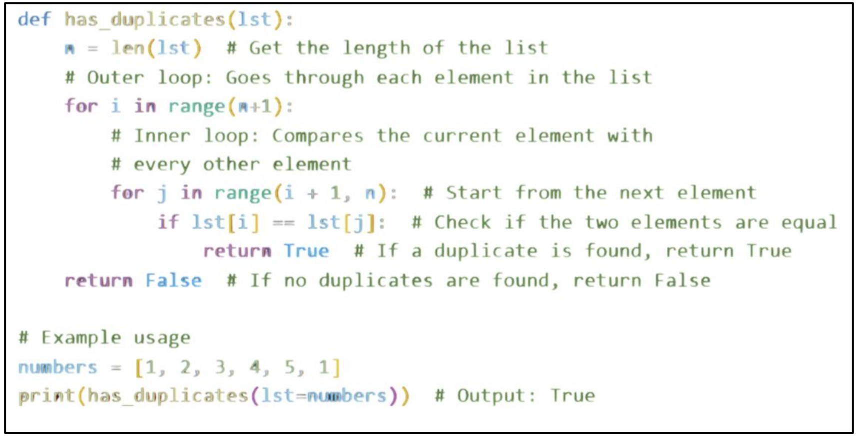
User: Give me a fast method in python to check if a list contains duplicated items.

Model response A:



```
def has_duplicates(lst):
    seen = set()
    for x in lst:
        if x in seen:
            return True
        seen.add(x)
    return False
```

Model response B:



```
def has_duplicates(lst):
    n = len(lst)  # Get the length of the list
    # Outer loop: Goes through each element in the list
    for i in range(n+1):
        # Inner loop: Compares the current element with
        # every other element
        for j in range(i + 1, n):  # Start from the next element
            if lst[i] == lst[j]:  # Check if the two elements are equal
                return True  # If a duplicate is found, return True
    return False  # If no duplicates are found, return False

# Example usage
numbers = [1, 2, 3, 4, 5, 1]
print(has_duplicates(lst=numbers))  # Output: True
```

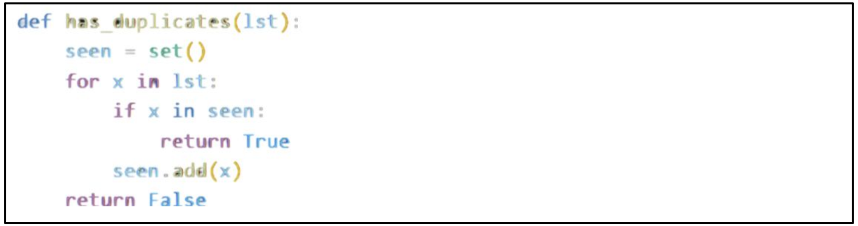
In this solution:

- The answer is comprehensive, demonstrating that aspects of correctness and informativeness of both responses were addressed in the respective input boxes.
- The depth of analysis is commendable, as the two approaches were compared in terms of their time complexity.
- The most important differences between the two model responses were identified: comments, test case, incorrect range in the outer FOR loop and the difference between the two algorithms (set() vs FOR loops).
- The text is free of spelling/grammar mistakes, well-structured, coherent and seamlessly articulate the most important findings of the comparison.

Bad Example

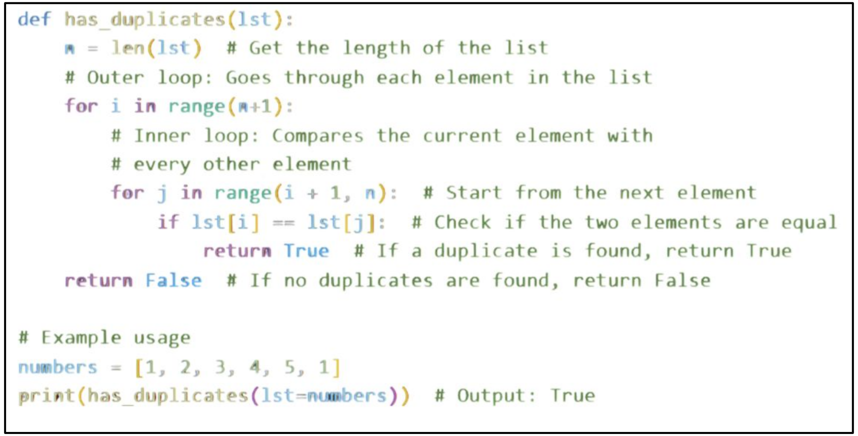
User: Give me a fast method in python to check if a list contains duplicated items.

Model response A:



```
def has_duplicates(lst):
    seen = set()
    for x in lst:
        if x in seen:
            return True
        seen.add(x)
    return False
```

Model response B:



```
def has_duplicates(lst):
    n = len(lst)  # Get the length of the list
    # Outer loop: Goes through each element in the list
    for i in range(n+1):
        # Inner loop: Compares the current element with
        # every other element
        for j in range(i + 1, n):  # Start from the next element
            if lst[i] == lst[j]:  # Check if the two elements are equal
                return True  # If a duplicate is found, return True
    return False  # If no duplicates are found, return False

# Example usage
numbers = [1, 2, 3, 4, 5, 1]
print(has_duplicates(lst=numbers))  # Output: True
```

Solution:

Correctness: Consider the factuality, logical flow, contextual relevance, and consistency.

Response A is good because set a fast function. Only use here 1 for and 1 if. Code is short.

Response B is also good because is return corrected result. FOR loops is correct.

Informativeness: Is the information provided relevant to the user's request? Is the information complete?

Response B. It has more comments.

Which one would you choose as the best possible model response? Explain why.

Response A is better than B because using set is the best solution.

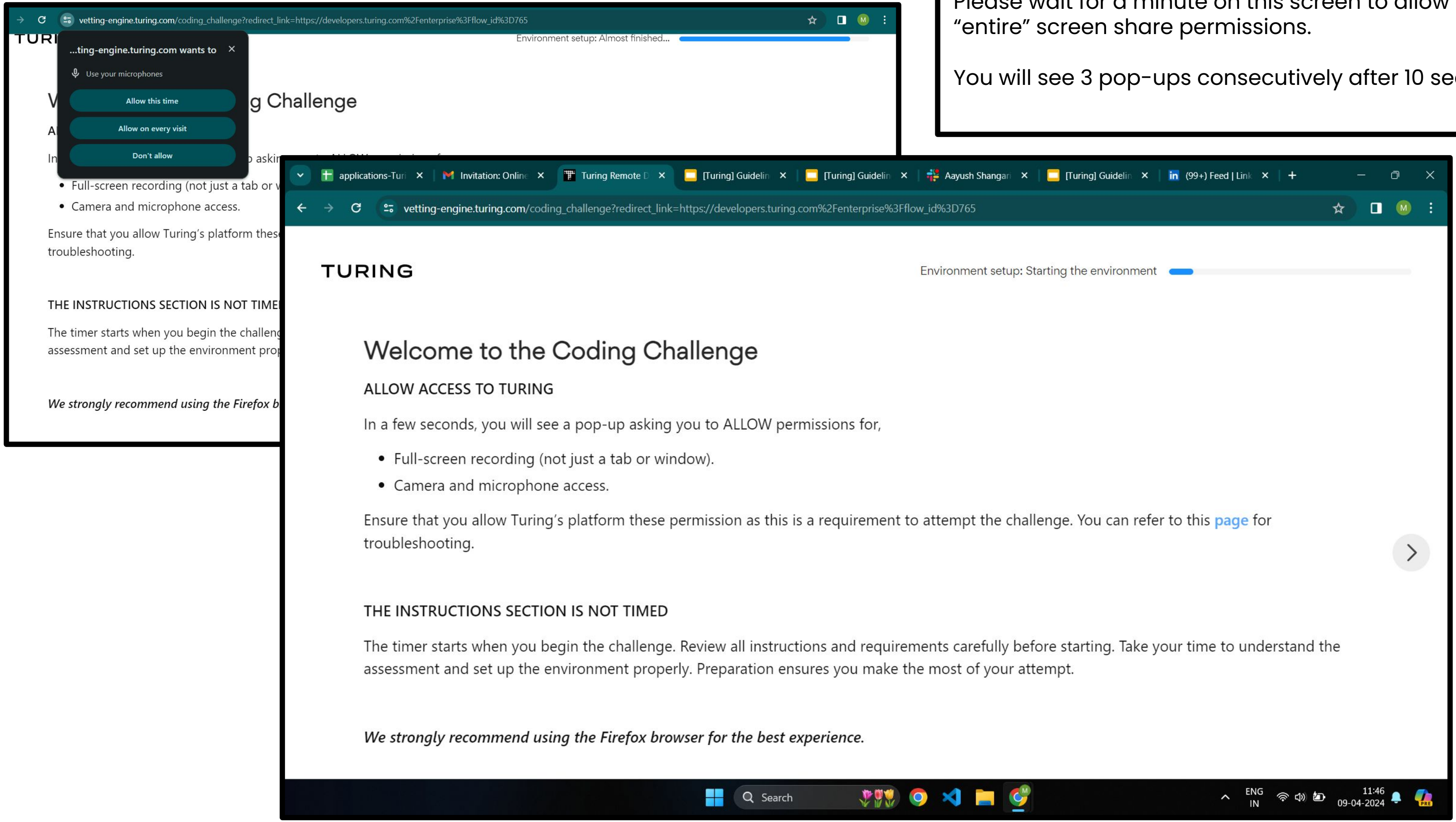
In this solution:

- The answer is too short lacking a comprehensive comparison of the two model responses.
- In "Informativeness", response A is not even discussed.
- The depth of analysis is too superficial.
- Some important differences are missing: comments, test case, incorrect range in outer FOR loop.
- Spelling/grammar mistakes are notorious.

This is not a good solution, even though model response A is indeed better than B, and the set() is a better approach.

Please wait for a minute on this screen to allow audio, video and “entire” screen share permissions.

You will see 3 pop-ups consecutively after 10 seconds each.


vetting-engine.turing.com/coding_challenge?redirect_link=https://developers.turing.com%2Fenterprise%3Fflow_id%3D765
Environment setup: Almost finished...
...ting-engine.turing.com wants to
Use your microphones
Allow this time
Allow on every visit
Don't allow
applications-Turi
Invitation: Online
Turing Remote D
[Turing] Guidelin
Aayush Shangari
(99+) Feed | Link
TURING
Environment setup: Starting the environment
Search
ENG
IN
11:46
09-04-2024

Welcome to the Coding Challenge

ALLOW ACCESS TO TURING

In a few seconds, you will see a pop-up asking you to ALLOW permissions for,

- Full-screen recording (not just a tab or window).
- Camera and microphone access.

Ensure that you allow Turing’s platform these permission as this is a requirement to attempt the challenge. You can refer to this page for troubleshooting.

THE INSTRUCTIONS SECTION IS NOT TIMED

The timer starts when you begin the challenge. Review all instructions and requirements carefully before starting. Take your time to understand the assessment and set up the environment properly. Preparation ensures you make the most of your attempt.

We strongly recommend using the Firefox browser for the best experience.

TURING

You’re good to start!

Go back to the assessment tab to begin.