Task 1

1. Manual Python Function (Sort by Key)

```
def sort_dicts_by_key(data, key):
    sorted_data = sorted(data, key=lambda x: x.get(key))
    return sorted_data
```

2. AI-Suggested (GitHub Copilot Style)

Copilot might suggest something like this:

```
def sort_list_of_dicts(lst, sort_key):
    return sorted(lst, key=lambda d: d[sort_key])
```

3. Comparison and Analysis (200 words)

Both implementations aim to sort a list of dictionaries by a given key. The manual version uses .get(key) within the lambda, which gracefully handles cases where the key might not be present, returning None and preventing a crash. In contrast, the Copilot version directly accesses the key (d[sort_key]), which can raise a KeyError if the key is missing in any dictionary.

From a readability standpoint, both are concise and easy to understand, but the manual version is more robust and safer in scenarios where the key presence is not guaranteed. If all dictionaries are guaranteed to contain the key, the Copilot version is slightly faster since it avoids the overhead of .get().

In terms of efficiency, the difference is negligible unless sorting a very large list. Performance-wise, both rely on Python's sorted() method, which uses Timsort (O(n log n)) under the hood.

Conclusion: The manual implementation is more reliable, especially in real-world data handling. The Copilot version is fine for controlled environments but risks runtime errors without key validation..