NAME:PAREENITA A.SHIRSATH B.E.A.I.&.D.S. ROLL.NO:57

BDA EXPERIMENT : 5(B)

```
def jaccard_distance(set1, set2):
    """
    Calculates the Jaccard distance between two sets.

    Args:
        set1: The first set.
        set2: The second set.

    Returns:
        The Jaccard distance as a float.
    """
    intersection = len(set1.intersection(set2))
    union = len(set1.union(set2))

    # Handle the case where both sets are empty to avoid division by zero
    if union == 0:
        return 0.0  # Jaccard distance is 0 if both sets are empty

    jaccard_similarity = intersection / union
    jaccard_dist = 1 - jaccard_similarity
    return jaccard_dist

# Example usage:
set_a = {"apple", "banana", "orange", "kiwi"}
set_b = {"banana", "grape", "kiwi", "pineapple"}

distance = jaccard_distance(set_a, set_b)
print(f"Jaccard distance between set_a and set_b: {distance}")


set_empty1 = set()
set_empty2 = set()
distance_empty = jaccard_distance(set_empty1, set_empty2)
print(f"Jaccard distance between two empty sets: {distance_empty}")
```

```
Jaccard distance between set_a and set_b: 0.6666666666666667
Jaccard distance between two empty sets: 0.0
```

```
def jaccard_distance(set1, set2):
    """
    Calculates the Jaccard distance between two sets.

    Args:
        set1: The first set.
        set2: The second set.

    Returns:
        The Jaccard distance as a float.
    """
    intersection = len(set1.intersection(set2))
    union = len(set1.union(set2))

    # Handle the case where both sets are empty to avoid division by zero
    if union == 0:
        return 0.0  # Jaccard distance is 0 if both sets are empty

    jaccard_similarity = intersection / union
    jaccard_dist = 1 - jaccard_similarity
    return jaccard_dist

# Example 1: Overlapping fruit sets
set_a = {"apple", "banana", "orange", "kiwi"}
set_b = {"banana", "grape", "kiwi", "pineapple"}
distance = jaccard_distance(set_a, set_b)
print(f"Example 1 - Jaccard distance between set_a and set_b: {distance:.4f}")

# Example 2: Completely disjoint sets
set_c = {"cat", "dog", "mouse"}
set_d = {"car", "bike", "plane"}
distance_disjoint = jaccard_distance(set_c, set_d)
```

```
print(f"Example 2 - Jaccard distance between disjoint sets: {distance_disjoint:.4f}")

# Example 3: Identical sets
set_e = {"red", "blue", "green"}
set_f = {"red", "blue", "green"}
distance_identical = jaccard_distance(set_e, set_f)
print(f"Example 3 - Jaccard distance between identical sets: {distance_identical:.4f}")

# Example 4: One empty, one non-empty set
set_g = set()
set_h = {"one", "two", "three"}
distance_one_empty = jaccard_distance(set_g, set_h)
print(f"Example 4 - Jaccard distance with one empty set: {distance_one_empty:.4f}")

# Example 5: Both empty sets
set_empty1 = set()
set_empty2 = set()
distance_empty = jaccard_distance(set_empty1, set_empty2)
print(f"Example 5 - Jaccard distance between two empty sets: {distance_empty:.4f}")
```

```
Example 1 - Jaccard distance between set_a and set_b: 0.6667
Example 2 - Jaccard distance between disjoint sets: 1.0000
Example 3 - Jaccard distance between identical sets: 0.0000
Example 4 - Jaccard distance with one empty set: 1.0000
Example 5 - Jaccard distance between two empty sets: 0.0000
```