### Programming Guide for “Hash Tables”

#### 1. HashTable Class Implementation

**Overview:** A HashTable is a data structure that stores data in an associative manner. In a hash table, data is stored in an array format, where each data value has its own unique index value.

**Key Components:** - **Hash Function:** Determines how to distribute entries across an array. - **Collision Handling:** Deals with two keys hashing to the same index.

**Pseudo Code for HashTable class:**

CLASS HashTable  
 FUNCTION \_\_init\_\_(size)  
 INITIALIZE array of given size  
  
 FUNCTION \_hash(key)  
 COMPUTE hash for the given key  
  
 FUNCTION put(key, value)  
 FIND index using \_hash(key)  
 STORE value at the computed index  
  
 FUNCTION get(key)  
 FIND index using \_hash(key)  
 RETURN value at the computed index  
  
 FUNCTION remove(key)  
 FIND index using \_hash(key)  
 REMOVE the value at the computed index  
END CLASS

**Implementation Tips:** - Use a Python list to simulate an array for storing key-value pairs. - The \_hash method should uniformly distribute keys. A simple modulo operation with the array size can be used. - Handle collisions using techniques like chaining (linked lists at each array index) or open addressing (finding the next empty slot).

#### 2. Anagram Check Function

**Overview:** An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

**Pseudo Code for is\_anagram:**

FUNCTION is\_anagram(s1, s2)  
 IF lengths of s1 and s2 are different  
 RETURN False  
 SORT s1 and s2  
 RETURN if sorted s1 equals sorted s2  
END FUNCTION

**Implementation Tips:** - The simplest way to check for anagrams is to sort both strings and compare them. - Sorting rearranges the characters so that if both strings contain the same characters, they will appear in the same order post-sorting. - This approach is effective for strings with a reasonable length. For very long strings, more complex methods might be more efficient.