**2.8 SEQUENCE OF CHARACTERS**

**AIM**

To return all strings in words that is a substring of another word. You can return the answer in any order

**ALGORITHM**

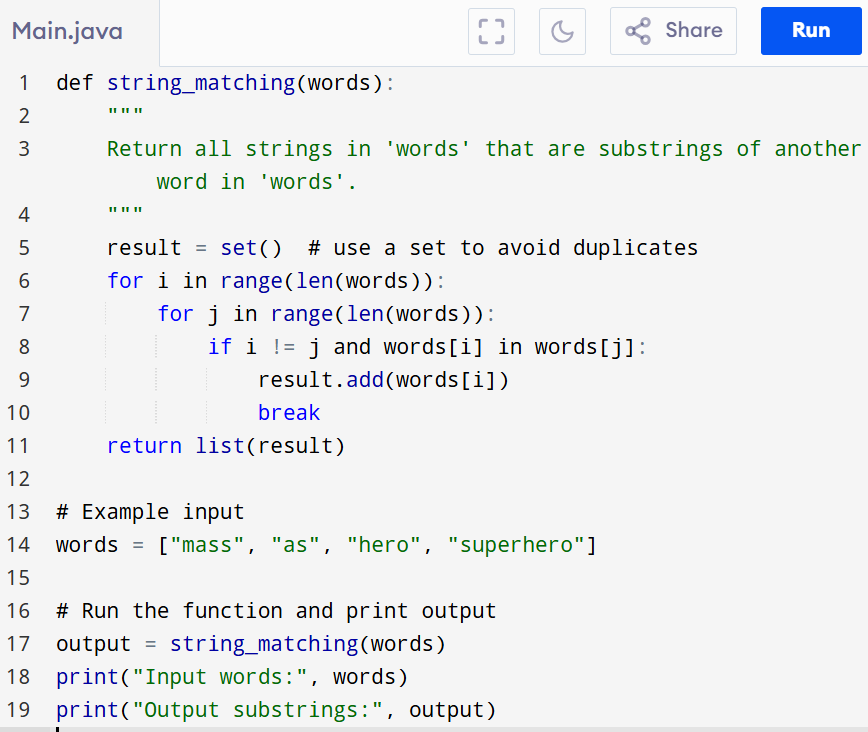
1. Iterate over each word in words.

2. For each word, check every other word in the list.

3. If the current word is found as a substring of any other word, add it to the result.

4. Return the list of such words.

**PROGRAM**



Input:

words = ["mass", "as", "hero", "superhero"]

Output:

A screenshot of a computer

AI-generated content may be incorrect.

**RESULT:**

Thus the program is successfully executed and the output is verified.

**PERFORMANCE ANALYSIS:**

**Time Complexity:**

* Sorting: O(n log n)
* Nested loops: For each word, check all longer words → O(n²)
* Substring check (word in other\_word): O(k \* k) in worst case
* Overall: **O(n² \* k²)**

**Space Complexity:**

* Output list: O(n) in the worst case (all words could be substrings)
* Additional space: O(1)