Question:4

CRYPTARITHMETIC PROBLEM

AIM

To solve the Cryptarithmetic problem using Python

ALGORITHM

1. is_mapping_possible():
    a. Generates all possible permutations of digits for the unique characters in the strings `arr` and string `S`.
    b. For each permutation, creates a mapping of characters to digits.
    c. Converts each string in `arr` into an integer using the character mapping and calculates their sum.
    d. Converts string `S` into an integer using the same character mapping.
    e. If the sum of integers formed by `arr` is equal to the integer formed by `S`, returns True.
    f. Otherwise, returns False.
2. sum_strings():
    a. Sums the integers formed by converting each string in `arr` using the given character mapping.
3. string_to_int():
    a. Converts a string into an integer using the given character mapping.
4. Example usage:
    a. Defines `arr` as ["SEND", "MORE"] and `S` as "MONEY".
    b. Calls `is_mapping_possible()` with `arr` and `S`.
    c. Prints "Yes" if it's possible to map integers to characters such that the sum of integers formed by `arr` is equal to the integer formed by `S`, otherwise prints "No".

CODE

```
from itertools import permutations

def is_mapping_possible(arr, S):
    unique_chars = set(''.join(arr) + S)
```

```
    if len(unique_chars) > 10:
        return False

    for perm in permutations(range(10), len(unique_chars)):
        char_map = {char: num for char, num in zip(unique_chars, perm)}
        if sum_strings(arr, char_map) == string_to_int(S, char_map):
            return True
    return False

def sum_strings(arr, char_map):
    total = 0
    for string in arr:
        total += string_to_int(string, char_map)
    return total

def string_to_int(string, char_map):
    num = 0
    for char in string:
        num = num * 10 + char_map[char]
    return num

# Test the function with the given input
arr = ["SEND", "MORE"]
S = "MONEY"
print("Output:", "Yes" if is_mapping_possible(arr, S) else "No")
```

OUTPUT