

PROGRAM-4

CRYPTO ARITHMETIC PROBLEM

AIM:-

To write and execute the python program for the crypto arithmetic program.

PROCEDURE:-

Import Required Modules:

- Import the permutations function from the itertools module.

Check Mapping Possibility:

- Define the is_mapping_possible function, which takes the list of strings arr and the target string S as input.
- Get the set of unique characters present in both the strings in arr and the target string S.
- If no valid mapping is found after checking all permutations, return False.

Convert String to Integer:

- Define the string_to_int function, which converts a string to an integer using a given character map.
- Initialize a variable num to store the integer representation.

Calculate Sum of Encoded Strings:

- Define the sum_strings function, which calculates the total sum of the numbers formed by encoding each string in arr using the character map.
- Initialize a variable total to store the sum..

Test the Function:

- Test the is_mapping_possible function with the given input arr and S.
- Print "Yes" if it's possible to map the characters in arr to integers such that their sum equals the string S, otherwise print "No".

CODING:-

```
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:-

A screenshot of the IDLE Shell 3.11.4 window. The window title is "IDLE Shell 3.11.4". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The shell area shows the following text: "Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32", "Type 'help', 'copyright', 'credits' or 'license()' for more information.", and a prompt ">>>". The user has entered "= RESTART: C:/Users/User/AppData/Local/Programs/Python/Python311/program 4.py" and the shell has responded with "Output: Yes". The prompt ">>>" is shown again on the next line.

```
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/User/AppData/Local/Programs/Python/Python311/program 4.py
Output: Yes
>>>
```

RESULT:-

Hence the program has been successfully executed and verified.