SCHOOL OF SCIENCE AND TECHNOLOGY, SINGAPORE
MATHEMATICS DEPARTMENT 2022
SECONDARY 4

Name: _____ **( )**

Class: S4-0_____ **Date:** _____

March Python Recalibration Test

Super Spy Training School (SSTS) has a hearing post which is able to pick up fragmented signal messages. After some analysis, it was discovered the original message, *OM*, can be extracted by using all the fragmented signals.

The signal fragments are received in the following order:

- All odd numbered fragments of *OM* are received first in ascending order. i.e. {1, 3, 5, …, last odd fragment},
- Followed by even numbered fragments of *OM* in ascending order. i.e. {0, 2, 4, 6, …, last even fragment}.

All signal fragments have the same number of characters.

The school has derived the following formula for the number of fragments that can be re-combined to form *OM*:

$$F = S - Ch + 1$$

where `F` is the total number of fragments
`S` is the length of *OM*
`Ch` is the number of characters in a signal fragment

SSTS wants a program that can rearrange the signal fragments to the original order they are found in *OM*.

It should have the following criteria:

- Take input, the signal fragments, one by one, as strings.
- Allow the user to continue to input until all signal fragments have been entered. The user should determine an end input string condition. e.g. `"!END"`
- Output all the signal fragments in the original order (i.e. {0, 1, 2, 3, …, last signal fragment}), in a single line each separated by a space.
- Output the length of *OM*.

A suitable input message must be used for each input. The inputs do not need to be validated.

1. Write the function for the given criteria. **[11]**

Save your program as **CODE**_<your class>_<index number>_<your name>**.py**
(e.g. “CODE_S409_08_John Tan.py”)

2. When your program is working, use the following test data to show your test result.

```
OMP
PUT
TIN
NGP
PLU
COM
MPU
UTI
ING
GPL
LUS
```

Take a screenshot of your result and save it as:

CODETEST_<your class>_<index number>_<your name> **[2]**
(e.g. "CODETEST_S409_08_John Tan")

Save your file in either **.jpg** or **.png** format.

3. Save your program as **CODE2**_<your class>_<index number>_<your name>**.py**
(e.g. "CODE2_S409_08_John Tan.py")

Each fragment may have `Ch - 1` overlapping characters.

Extend your program to recreate the original signal message in a single string using all the signal fragments. Output the recreated original message. **[4]**

Save your program.

4. Save your program as **CODE3**_<your class>_<index number>_<your name>**.py**
(e.g. "CODE3_S409_08_John Tan.py")

The signal fragments are now randomly ordered. Write a user-defined function `FindEnds(F)`, where the parameter `F` is a list of the signal fragments, that will return the start character and end character of the original message. **[3]**

Save your program.

Example:

```
SSS
SST
STX
STX
TXS
TXT
XSS
XTT
```

Returns: `ST`