Lettuce 20XX Programming Competition

Problem 99: The Information Security Crucible

Congratulations! Today marks the day that you’ve graduated with High School! Your sure that this day can’t get any better, and then you receive an email from the college that you plan to go to.

The CCT has sent you an email with the subject line “1^{st} = APPLE”. After that weird subject line, you find out that you can’t read the email! There three things that you know from looking at the email: the first word must be apple, you notice that all the words are made from uppercase letters, and that the entire email seems to be just have the letters shifted in the alphabet (and you can realize this because you are a young info sec major and it reminds you of ROT13!).

Being the overachieving high school student (soon to be information security major), you decide that you want to write a program to help you decode the entire string of words for you!

Input

There will be N lines ($0 \leq N \leq 16$) of the email, of which you need to decode.

Each word can be separated by either “|”,” “, or “_”. Every word is in capital letters, and there are no numbers or other characters.

Sample input:

```
HWWSL OLSSV AOLYL ZABKLUA DL HYL WSLHZLK AV PUMVYT FVB AOHA FVB OHCL DVU
ZJOVSHYZOPW TVULF PU VYKLY AV HPK FVB PU FVBY PUMVYTHAPVU ZLJBYPAF THQVY
```

Output

Each data set should produce one line of output indicating the cipher that should be used to decode the email, in the order of A → Z (index 0 is what A is, index 1 is what B is, and so forth).

If there are no lines, output “Cipher not computable”.

Sample output:

```
[‘H’, ‘I’, ‘J’, ‘K’, ‘L’, ‘M’, ‘N’, ‘O’, ‘P’, ‘Q’, ‘R’, ‘S’, ‘T’, ‘U’ , ‘V’, ‘W’,
‘X’, ‘Y’, ‘Z’, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’]
```

-THIS PAGE IS INTENTIONALLY LEFT BLANK-

Answer Code

```
alpha=["A","B","C","D","E","F","G","H","I","J","K","L","M","N","O","P","Q","R","S","T","U","V","W","X","Y","Z"]
try:
  line = input()
except:
  print("Cipher not computable")
shouldBeA = line[0]
move = ord(shouldBeA) - 65
if move == 0:
  print(alpha)
else:
  alpha.extend(alpha[0:move])
  print(alpha[move:26+move])
```

Secret Inputs and Outputs

Because the possible inputs are so limited (I can start with A and shift all the way to Z), I would test 27 cases. I would only input the first word as well, because technically that is all that they need to solve the word, and that there can be n line of inputs that they don't need. So, start with a shift of 0 and shift all the way to having Z be A. After that, the 27th test would be an empty file, to see if they output "Cipher not computable". All 27 test cases are in their own file.