

1 Week 5 Exercises

1.1 List manipulation

1.1.1 List slicing

Using the **range()** function, create a list containing all numbers less than 50. Then use list slicing (not the % operator!) to:

- Print all even numbers less than 50
- Print all multiples of 5 less than 50

Extension Creating a list of multiples this way is wasteful, as we're throwing away a bunch of numbers (e.g. 1,2,3,4;6,7,8,9... are all useless). Using a **for** loop, create a list of multiples of 5 directly.

1.1.2 Forced Anagrams

(Skip this if you did the Pascal's Triangle reading week challenge)

Take a word from the user, and then print a partially anagrammed version of the word (i.e. don't repeat the middle character).

- E.g. "fish"-> "fishsif"

1.2 "Fun" with strings

1.2.1 Goat Latin

Implement goat latin (a made up version of english [this is a facebook interview question!]), such that a user can input a normal english sentence, and your program returns the goatified version:

1. If a word begins with a vowel, append "ma" to the end of the word.
 - "apple" becomes "applema"
2. If a word begins with a consonant (i.e. not a vowel), remove the first letter and append it to the end, then add "ma".
 - "goat" becomes "oatgma"
3. Add one letter 'a' to the end of each word per its word index in the sentence, starting with 1.
 - Example, the first word gets "a" added to the end, the second word gets "aa" added to the end and so on.

Note: you may find the following list useful:

- vowels = ['a', 'e', 'i', 'o', 'u']

Extension Instead of adding unlimited extra 'a's with increasing word index, add one, then two, then three, and then one again, etc.

- E.g.: "word1 word2 word3 word4 word5..." -> "aword1 aaword2 aaaword3 aword4 aa-word5..."

1.3 For loops

1.3.1 Hip to be Square

Take a number N from the user, and print an NxN square using characters 'X'

Extension 1

Implement this solution using string manipulation and only one for loop (if you already did this, great!)

Extension 2

Print a "hollow" square (i.e. only the edges are X")

1.3.2 Pythagorean triples - Three times the fun!

Pythagorean triples are sets of three integers which satisfy pythagoras' theorem:

- $a^2 + b^2 = c^2$

For example, the first pythagorean triple is (3, 4, 5).

Find all pythagorean triples where $c < 100$ and print these out (ignoring duplicates in a and b), formatted as:

(3, 4, 5), (a2, b2, c2), (a3, b3, c3) ...