12. Reverse the below list without using any inbuilt keywords (like reverse() or [::-1])

List = ["cat","tiger","lion", "zebra" , "crocodile", "snack"]

Day2

1. Define a function generate_n_chars() that takes an integer n and a character c and returns a string, n characters long, consisting only of c:s. For example, generate_n_chars(5,"x") should return the string "xxxxx". (Python is unusual in that you can actually write an expression 5 * "x" that will evaluate to "xxxxx". For the sake of the exercise you should ignore that the problem can be solved in this manner.)

2. The function max() from exercise 1) and the function max_of_three() from exercise 2) will only work for two and three numbers, respectively. But suppose we have a much larger number of numbers, or suppose we cannot tell in advance how many they are? Write a function max_in_list() that takes a list of numbers and returns the largest one.

3. Write a program that maps a list of words into a list of integers representing the lengths of the correponding words.

4. Write a function find_longest_word() that takes a list of words and returns the length of the longest one. Modify the same to do with lambda expression.

5. Write a function filter_long_words() that takes a list of words and an integer n and returns the list of words that are longer than n. Modify the same to do with lambda expression.

6. Write a version of a palindrome recognizer that also accepts phrase palindromes such as "Go hang a salami I'm a lasagna hog.", "Was it a rat I saw?", "Step on no pets", "Sit on a potato pan, Otis", "Lisa Bonet ate no basil", "Satan, oscillate my metallic sonatas", "I roamed under it as a tired nude Maori", "Rise to vote sir", or the exclamation "Dammit, I'm mad!". Note that punctuation, capitalization, and spacing are usually ignored.

7. A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to check a sentence to see if it is a pangram or not.

8. Represent a small bilingual lexicon as a Python dictionary in the following fashion {"merry":"god", "christmas":"jul", "and":"och", "happy":gott", "new":"nytt", "year":"år"} and use it to translate your Christmas cards from English into Swedish. That is, write a function translate() that takes a list of English words and returns a list of Swedish words.

9. Write a function char_freq() that takes a string and builds a frequency listing of the characters contained in it. Represent the frequency listing as a Python dictionary. Try it with something likechar_freq("abbabcbdbabdbdbababababcbcbab").
10. Create a module called mathematics.py and provide subroutines (should be defined generally and should work for any number of arguments) such as:

Add Sub

Sort the values Max Sort