

CSC1015F Assignment 9

Recursion

Assignment Instructions

This assignment involves constructing Python functions that use recursion. You must **NOT** use loop constructs (such as 'for' and 'while') in your solutions.

Assessment

Your code will be automatically marked. Say that there are N trials for a question. The first $(N-1)$ trials will check that your code functions correctly by executing it on test inputs. The N th is a penalty test. It scans the code for evidence of the use of iteration or the use of list/string reversal expressions. If it finds evidence, then it deducts the marks for the question.

In some cases, the penalty test will report a false positive. For instance, it thinks you're using loops but you are not, you simply have a variable name containing the word 'for', e.g. 'former', 'afford'.

Furthermore, your solutions to this assignment will be evaluated for correctness and for the following qualities:

- Documentation
 - Use of comments at the top of your code to identify program purpose, author and date.
 - Use of comments within your code to explain each non-obvious functional unit of code.
- General style/readability
 - The use of meaningful names for variables and functions.
- Algorithmic qualities
 - Efficiency, simplicity

These criteria will be manually assessed by a tutor and commented upon. Up to 10 marks will be deducted for deficiencies.

Question one [30 marks]

Write a program called 'pairs.py' that uses a recursive function to count the number of pairs of consecutive characters in a string. Pairs of characters cannot overlap. You MUST NOT use any form of loop in your program!

(NOTE: the input from the user is shown in **bold font**.)

Sample I/O:

Enter a message:

Tebello Sello is bleeding, and needs medical attention

Number of pairs: 5

You may NOT use iteration, or a string slice expression (or any other technique) to reverse the string without using recursion!

Question three [30 marks]

Write a program called 'encrypt.py' that uses a recursive function to encrypt a message by converting all lowercase characters to the next character (with z transformed to a).

You MUST NOT use any form of loop in your program! (Yep, we've definitely mentioned this.)

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter a message:

hello world

Encrypted message:

ifmmp xpsme

Question Three (Palindrome Primes) [40 marks]

Write a program called 'palindromeprimes.py' that uses recursive functions to find all palindromic primes between two integers N , M , supplied as input. (start and end points are included).

- A palindrome number is a number that reads the same from the front and the back. Examples are: 212, 44, 9009, 4567654.
- To calculate whether a number is a palindrome or not, incorporate your answer to question 1.
- A prime number is a number greater than 1 that is only divisible by 1 and itself. Examples are: 3, 11, 313.

Some examples of palindromic primes are: 11, 191, 313.

You may assume it's always the case that $N > 1$, and that $N \leq M$.

Version: 19/04/2023 22:02

*You **MUST NOT** use any form of loop in your program! (Yes, we're pretty obsessive about this.)*

Add the following lines at the top of your program to increase the amount of recursion that Python will allow:

```
import sys
sys.setrecursionlimit (30000)
```

Sample IO (The input from the user is shown in **bold font** – do not program this):

Enter the starting point N:

200

Enter the ending point M:

800

The palindromic primes are:

313

353

373

383

727

757

787

797

Submission

Create and submit a Zip file called 'ABCXYZ123.zip' (where ABCXYZ123 is YOUR student number) containing `pairs.py`, `encrypt.py` and `palindromeprimes.py`.