INFS1201-Computer Program

Lab 11

Submit all exercises!

Always start your code by the following comments:

##

#firstName lastName - StudentID

# Lab 11

#Exercise XXX

## Exercise 1

An arithmetico-geometric sequence is a sequence which combines arithmetic and geometric characteristics. It is defined as follows:

u0=initial

ui=ratio\*ui-1 + const

A screenshot of a computer

Description automatically generated

Write a function arithmeticGeometric(index, const, ratio, initial) which recursively computes the term index (uindex) of the sequence defined above.

As before you can test your function directly in the console as follows:

>>> arithmeticGeometric(5,2,1,0)

10

>>> arithmeticGeometric(5,0,2,1)

32

>>> arithmeticGeometric(5,2,2,1)

94

## Exercise 2

The Collatz sequence starts with a0 a positive integer larger than 1. Then the following terms of the sequence are defined as:

|  |  |
| --- | --- |
| Collatz Problem -- from Wolfram MathWorld  The sequence stops when an=1.  The Collatz conjecture states that, regardless of the starting number u0 the sequence always ends up reaching 1. Can you prove it? If you can, you might become rich! <https://en.wikipedia.org/wiki/Collatz_conjecture>  As proving it might end up being difficult, you are simply requested to write a function collatz(init) that **recursively** computes the following items of the sequence starting from init until a term becomes 1. Check with a few numbers that the sequence always terminates! | A screenshot of a white sheet with black text  Description automatically generated |

Sample run:

>>> collatz(5)

[5, 16, 8, 4, 2, 1]

>>> collatz(11)

[11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1]

## Exercise 3

Write a program that asks the user to enter 6 test scores and count how many of the test scores are passes. RULES: You may not use any loops! You must define the following two recursive functions:

def input\_scores(n)

This function will ask the user for n scores and return the scores as a list.

def count\_passes(scores):

This function will return the number of passes in the list of scores. You are permitted to use list slicing operations.

Enter score: 79

Enter score: 60

Enter score: 100

Enter score: 45

Enter score: 70

Enter score: 50

4

## Exercise 4

Based on a previous lab exercise, whereby an instructor wanted to find the answer key that provides the best average for his students.

After careful consideration, the instructor realized that this method could end up with very bad grades for some students. So, he is changing is optimization: he wants to find the answer key that will result in the highest minimum score for his students. In case many solutions give the same minimum score, the will chose among this solution the one that gives the highest average.

For example, if we have the following list of answers by 6 students:

attempts=['TTFTTFFTFT','FFFTTFFTFT','TTFFFFFTFT','TTFFFFFTFT','TTFTTTTTFT','TTTTTTTTTT']

Then the output of the program should be the following:

Arranged 'correct' answers: TTFTTFFTTT

Arranged scores: [9, 7, 7, 7, 7, 7]

Best min grade: 7