COMP1001 Additional programming exercises

1. Write a function that accepts a list of comparable values and returns the maximum of the values in the list. Try your function with [2,4,5,9,0,1,3] and ['apple','lemon','Panda','zebra','tiger'].

def com(list):

big = list[0]

for i in range(len(list)):

if list[i] >= big:

big = list[i]

print(big)

#这个程序我们学会了如何连续处理一列数，并保留一个最大的参数big

1. Write a Python program that will generate the following output.

6 7 4

6 7 5

6 7 6

6 7 4

6 7 5

6 7 6

6 7 4

6 7 5

6 7 6

8 7 4

8 7 5

8 7 6

8 7 4

8 7 5

8 7 6

8 7 4

8 7 5

8 7 6

for i in [6,8]:

for j in [7,7,7]:

for k in [4,5,6]:

print(i,j,k)

#这个程序告诉我们python能简单到这种程度

#还有就是，连续叠加for的输出效果。

1. Write a program that will accept a sentence from user. The program will return the number of words. To make it simple, we assume there are no punctuations in the sentence.

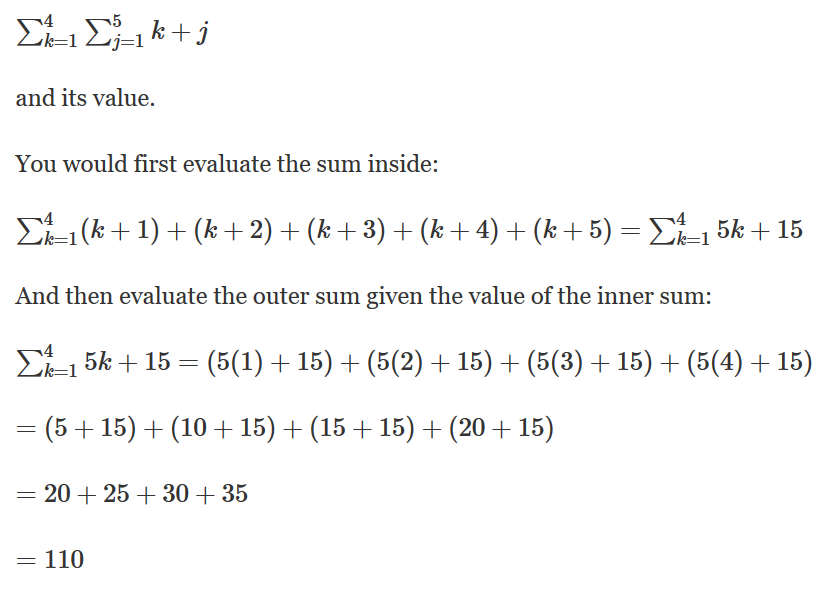
sen = input('A sentence:')

sen = sen.split(' ')

print(len(sen))

#这个程序告诉我们如何熟练运用split分离字符串

1. Consider the double summations below.



Write a Python program that will generate the result of the double summations.

k, j = eval(input('Please input k and j:'))

a, b = eval(input('Please input a and b:'))

#这是一个蛮简单，但没有表述清楚的数学问题。

1. A palindrome is a word, number, phrase, or other sequence of characters which reads the same backward as forward, such as madam or racecar or the number 10201. Write a Python function that will accept a string. It returns true if the input string is a palindrome and false otherwise.

def palindrorm(str):

output = 'yes'

for i in range(len(str)):

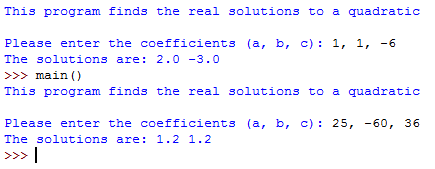
if str[i] != str[-1-i]:

output = 'no'

return(output)

# 这个程序告诉我们，解决问题的思路和算法，比如何编程更重要。

1. Write a Python program to find the real solutions of the quadratic equation . The roots can be found by the following formula:



import math

def main():

print('This program finds the real solution to a quadratic')

print()

a,b,c = eval(input('Please enter the cofficients (a, b, c): '))

x1 = (-b + math.sqrt(b\*b-4\*a\*c))/(2\*a)

x2 = (-b - math.sqrt(b\*b-4\*a\*c))/(2\*a)

print('The solution are: '+ str(x1) + ' '+ str(x2) )

# 这个程序告诉我们如何引入math函数做一些厉害的计算。

1. Write a function that accepts a string of words separated by space and returns a list of the words sorted lexicographically. Input some string of words to test your function. You cannot use any methods/built-in functions that can sort a list.

#还和我来词典顺序？早怎么不出这个题！

1. (Exam 2017/2018) Write a program that uses **brute force approach** to solve a classic ancient Chinese puzzle: “We count 35 heads and 94 legs among the chickens and rabbits in a farm. How many rabbits and how many chickens do we have?” The program should display the number of rabbits and chickens respectively. If you obtain the answer by solving the two simultaneous equations, you will receive no marks.

def main():

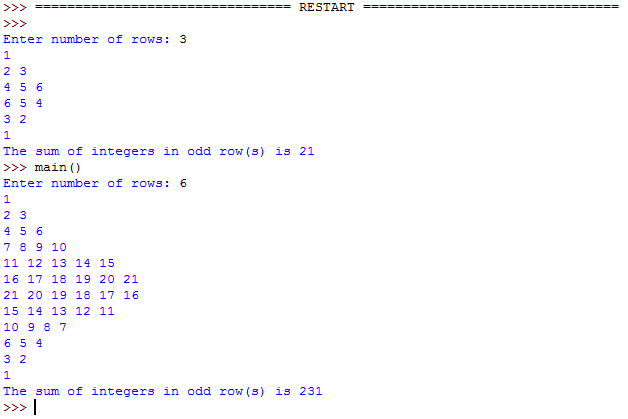
for chi in range(35):

if chi\*2 + (35-chi)\*4 == 94:

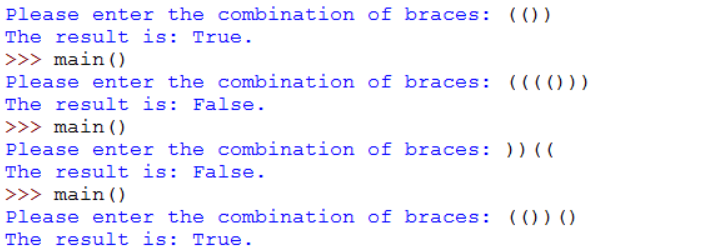
print('There are '+str(chi)+' chickens and '+str(35-chi)+ ' rabbits.')

1. (Exam 2017/2018) Floyd's triangle is a [right-angled](https://en.wikipedia.org/wiki/Right_angle) [triangular](https://en.wikipedia.org/wiki/Triangle) array of [positive](https://en.wikipedia.org/wiki/Natural_number) integers. It is defined by filling the rows of the triangle with consecutive integers, starting with a 1 in the top left corner. For example,

Write a program that accepts the number of rows for the Floyd's triangle and prints out the triangle. Immediately below the Floyd's triangle prints an image of the triangle but the integers on each row are reversed. After that, print the sum of the integers of odd-numbered rows (i.e., 1st row, 3rd row, 5th row and so on). Below are the sample inputs and outputs:



1. (Exam 2016-17) Write a Python program that will ask user for a combination of “(” and “)” and return True if the braces match in nested or separate pairs and False otherwise. Sample outputs below:



1. (Exam 2016-17) A Pascal’s Triangle, shown in Figure 1 for 7 levels, is a famous mathematical structure. A Pascal’s Triangle of 0 level simply consists of 1. For one level, it consists of 1 and 1. For two levels and above, an algorithm for computing the triangle is shown in Figure 2. Given row *i,* an inner element of row *i* + 1 is given by adding two adjacent numbers on row *i*.

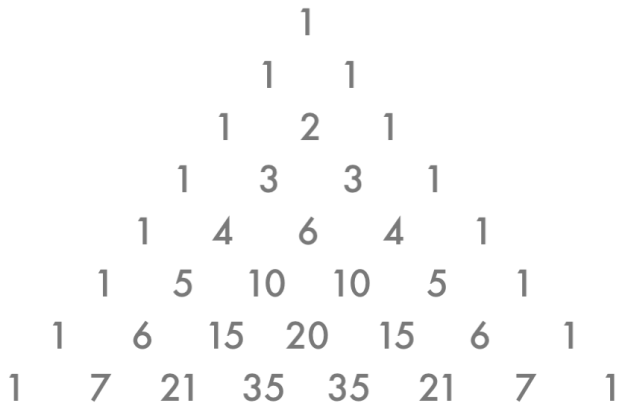
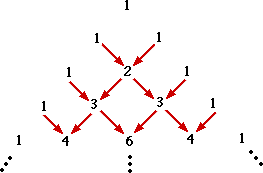
 

Figure 1: A Pascal’s Triangle of 7 levels. Figure 2: An algorithm for computing a

Pascal’s Triangle for 4 levels.

**Implement a function named pascalTri(n) that will print out the rows of numbers for a Pascal’s Triangle of *n* levels like the one below (but not the ones in Figures 1 and 2).** Include a main() that will print out a 10-level Pascal’s Triangle as depicted below.

