CS100 Tutorial 10

Hands-on small projects

Task 1

- Implement a function SumAll in python
  - SumAll takes a number n as input,
    1. if n is a positive integer, returns 1+2+…+n
    2. if n is not a positive integer, then raises the exception AssertionError with a message ”The input should be a positive integer”

Task 1: Testcases

```
print(SumAll(100))
>>>5050

print(SumAll(0))
>>>Traceback (most recent call last):
  ...
AssertionError: The input should be a positive integer

print(SumAll("abc"))
>>>Traceback (most recent call last):
  ...
AssertionError: The input should be a positive integer
```

Task 2

- Implement an optimal function Combination in python (without importing any module)
  - Combination takes two numbers n and i as inputs,
    1. if either n or i is not a positive integer, or i<n then raises the exception AssertionError with a message ” n and i must be positive integers and n must be larger than or equal to i.”
    2. else returns the result of

$$\frac{n!}{i! \times (n-i)!}$$

Task 2: Testcases

print(Combination(3, 2))
>>>3

print(Combination(50, 25))
>>>126410606437752

print(Combination(3, 0))
>>>Traceback (most recent call last):
...
AssertionError: n and i must be positive integers and n must be larger than or equal to i.

Task 3

- Implement an optimal function GetFib in python (without importing any module)
  - GetFib takes a numbers n as input,
    1. if n is a negative integer, then raises the exception AssertionError with a message ” n cannot be a negative integer.”
    2. else returns the first number in Fibonacci sequence that is greater than n

Task 3: Testcases

print(GetFib(50))
>>>55
print(GetFib(500))
>>> 610

Task 4

- Implement a function GuessNumGame in python
  - GuessNumGame takes two positive integers maxValue and maxTimes as inputs,
    1. Generate a random integer v between 0 and MaxValue including 0 and MaxValue
    2. Iteratively read an input integer v' from console until v==v' or the number of input times exceeds maxTimes
       - if v==v', then output "Congratulation"
       - if v>v', then output "Too smaller"
       - if v<v', then output "Too larger"

Task 4: Hint

- **To generate a random integer between m and n, we can use to following code:**

```
from random import randint  # import the function randint from the module random
value = randint(m,n) # generate a random integer value between m and n
```

Task 4: Testcases

GuessNumGame(6,3)

Input/Output

Start to GUESS:4
Too larger
Guess again:2
Congratulations!

GuessNumGame(10,3)

Input/Output

Start to GUESS:5
Too smaller
Guess again:7
Too smaller
Guess again:9
Too smaller
Game over. FAIL.
The value is 10