Seminar 2 Preparation

Please note that the Jupyter Notebook environment is available in Codio for you to carry out these activities. This week there are **two** programming exercises that will help you understand two valuable language concepts – **recursion and regex**.

Recursion

One of the classic programming problems that is often solved by recursion is the towers of Hanoi problem. A good explanation and walkthrough are provided by Cormen & Balkcom (n.d.) - the link is in the reading list. (the code they used for their visual example is provided on their website as well).

- Read the explanation, study the code and then create your own version using Python (if you want to make it more interesting you can use asterisks to represent the disks). Create a version that asks for the number of disks and then executes the moves, and then finally displays the number of moves executed.

```
#runcount set initially as zero because of no moves done
runcount = 0


#define function for tower of Hanoi
def TowerOfHanoi(n, origin_peg, destination_peg, auxilary_peg):


    #if number of disk is greater than zero the diskd will be moved
    if n > 0:
        global runcount
        runcount += 1
        TowerOfHanoi(n - 1, origin_peg, auxilary_peg, destination_peg)
        print("Move disk", n, "from peg", origin_peg, "to peg", destination_peg)
        TowerOfHanoi(n - 1, auxilary_peg, destination_peg, origin_peg)


#ask user how many disks has the tower of hanoi
n = int(input("How many disk has the tower of Hanoi?"))
TowerOfHanoi(n, 'A', 'C', 'B')
```

#it prints the moves and the number of moves necessary to move the disks

#from the origin peg to the destination peg

print (runcount)

here is possible to see the outcome of having an tower of hanoi with 3 or 4 disks and at the end of the move list it appears the total number of moves required.

```
Python 3.8.8 (default, Apr 13 2021, 12:59:45)
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/robertocappiello/Documents/University of Essex/Secure Software Development/
Codio exercises/Seminar 2/Recursion/Tower of Hanoi.py', wdir='/Users/robertocappiello/Documents/
University of Essex/Secure Software Development/Codio exercises/Seminar 2/Recursion')

How many disk has the tower of Hanoi?3
Move disk 1 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 3 from peg A to peg C
Move disk 1 from peg B to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C
7

In [2]: runfile('/Users/robertocappiello/Documents/University of Essex/Secure Software Development/
Codio exercises/Seminar 2/Recursion/Tower of Hanoi.py', wdir='/Users/robertocappiello/Documents/
University of Essex/Secure Software Development/Codio exercises/Seminar 2/Recursion')

How many disk has the tower of Hanoi?4
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 3 from peg A to peg B
Move disk 1 from peg C to peg A
Move disk 2 from peg C to peg B
Move disk 1 from peg A to peg B
Move disk 4 from peg A to peg C
Move disk 1 from peg B to peg C
Move disk 2 from peg B to peg A
Move disk 1 from peg C to peg A
Move disk 3 from peg B to peg C
Move disk 1 from peg A to peg B
Move disk 2 from peg A to peg C
Move disk 1 from peg B to peg C
15

In [3]:
```

- What is the (theoretical) maximum number of disks that your program can move without generating an error?

The time complexity for the solution of Tower of Hanoi is O(2^n), where n is the number of discs. If the program uses more memory space than the stack size then stack overflow will occur and can result in a program crash.

- What limits the number of iterations? What is the implication for application and system security?