

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.

Towers of Hanoi Programme

This is my code and output for the Towers of Hanoi, this allows the user to input the amount of discs and outputs the amount of moves.

Code

```
# Towers of Hanoi Recursive Function

# This will output the disc moves to the pegs
def Hanoi(n , source, peg, auxiliary):
    if n==1:
        print ("Move disc 1 from peg",source,"to peg",peg)
        return
    Hanoi(n-1, source, auxiliary, peg)
    print ("Move disc",n,"from peg",source,"to peg",peg)
    Hanoi(n-1, auxiliary, peg, source)

# n is the number of discs in the tower and will determine the moves to solve it
# There should be a minimum of 3, this would solve it in 7 moves, more discs mean more moves are required
n = int(input("Enter number of discs: "))
# This will output the amount of moves
d = (2*n)-1
print("\nTotal Moves " + str(d))

# The pegs are numbered 1, 2, and 3
Hanoi(n, '1', '2', '3')
```

Output (using 3 discs)

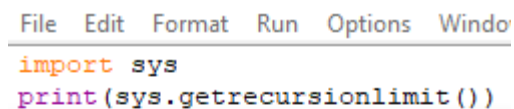
```
Enter number of discs: 3

Total Moves 7
Move disc 1 from peg 1 to peg 2
Move disc 2 from peg 1 to peg 3
Move disc 1 from peg 2 to peg 3
Move disc 3 from peg 1 to peg 2
Move disc 1 from peg 3 to peg 1
Move disc 2 from peg 3 to peg 2
Move disc 1 from peg 1 to peg 2
```


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

By default the highest number of discs is 1000, as that is the maximum recursion allowed in Python. This can be checked using `import sys`
`print(sys.getrecursionlimit())` and increased using the `sys.setrecursionlimit(n)` function.

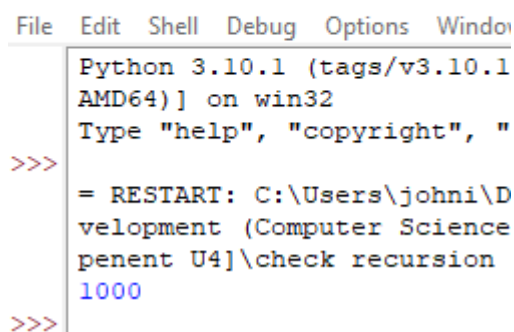
```
print(sys.getrecursionlimit())
```



```
File Edit Format Run Options Window
import sys
print(sys.getrecursionlimit())
```



IDLE Shell 3.10.1



```
File Edit Shell Debug Options Window
Python 3.10.1 (tags/v3.10.1
AMD64) on win32
Type "help", "copyright", "
>>>
= RESTART: C:\Users\johni\D
velopment (Computer Science
penent U4]\check recursion
1000
>>>
```

What limits the number of iterations?

The Python interpreter limits the amount of iterations at 1000, changing this is not advised as Python stack frames are big.

What is the implication for application and system security?

In terms of application and system security recursion can potentially be used to take advantage of vulnerabilities by causing a stack overflow. This could allow them to run their code on the application.

<https://www.pythoncentral.io/resetting-the-recursion-limit/>

[https://rollbar.com/blog/python-recursionerror/#:~:text=Increasing%20the%20recursion%20limit%3A%20Python,setrecursionlimit\(n\)%20function.](https://rollbar.com/blog/python-recursionerror/#:~:text=Increasing%20the%20recursion%20limit%3A%20Python,setrecursionlimit(n)%20function.)

[python - What is the maximum recursion depth, and how to increase it? - Stack Overflow](#)