

Seminar 2 Preparation - Recursion

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.

Code: (Tower of Hanoi.py file included in the e-portfolio)

```
#code from https://www.geeksforgeeks.org/python-program-for-tower-of-hanoi/
# Recursive Python function to solve the tower of hanoi

# modified to accept an input and print the number of move

def TowerOfHanoi(n , source, destination, auxiliary):
    global moves
    moves += 1

    if n==1:
        print ("Move disk 1 from source",source,"to destination",destination)
        return

    TowerOfHanoi(n-1, source, auxiliary, destination)
    print ("Move disk",n,"from source",source,"to destination",destination)
    TowerOfHanoi(n-1, auxiliary, destination, source)

# ask for number of disks
while True:
    try:
        n = int(input('please enter the number of disks:\n'))
        break

    except:
        continue

source = [i + 1 for i in reversed(range(n))]
destination = []
auxiliary = []

print(source, destination, auxiliary)
moves = 0

TowerOfHanoi(n, 'A', 'B', 'C')
print("Number of moves: ", moves)
```

Output:

```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\UOE\SSD\eportfolio\Unit4\Tower of Hanoi.py =====
please enter the number of disks:
5
[5, 4, 3, 2, 1] [] []
Move disk 1 from source A to destination B
Move disk 2 from source A to destination C
Move disk 1 from source B to destination C
Move disk 3 from source A to destination B
Move disk 1 from source C to destination A
Move disk 2 from source C to destination B
Move disk 1 from source A to destination B
Move disk 4 from source A to destination C
Move disk 1 from source B to destination C
Move disk 2 from source B to destination A
Move disk 1 from source C to destination A
Move disk 3 from source B to destination C
Move disk 1 from source A to destination B
Move disk 2 from source A to destination C
Move disk 1 from source B to destination C
Move disk 5 from source A to destination B
Move disk 1 from source C to destination A
Move disk 2 from source C to destination B
Move disk 1 from source A to destination B
Move disk 3 from source C to destination A
Move disk 1 from source B to destination C
Move disk 2 from source B to destination A
Move disk 1 from source C to destination A
Move disk 4 from source C to destination B
Move disk 1 from source A to destination B
Move disk 2 from source A to destination C
Move disk 1 from source B to destination C
Move disk 3 from source A to destination B
Move disk 1 from source C to destination A
Move disk 2 from source C to destination B
Move disk 1 from source A to destination B
Number of moves:  31
>>> |
```

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

The maximum number of disks that cab moved without generating an error will depend on the computer resources.

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

The number of iterations may be limited by the amount memory available. The applications may hang or crash and this might cause data loss or other issues to the system files.