

```
#1. Python program to solve Towers of Hanoi problem.

def hanoi(disks, src, aux, tar):
    if(disks == 1):
        print("Move disk 1 from rod {} to rod {}".format(src, tar))
        return
    hanoi(disks - 1, src, tar, aux)
    print("Move disk {} from rod {} to rod {}".format(disks, src, tar))
    hanoi(disks - 1, aux, src, tar)

disks = int(input("Enter the number of disks: "))
hanoi(disks, "A", "B", "C")
```

Output:

Enter the number of disks: 7

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 4 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 3 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 5 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 3 from rod B to rod A.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 4 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 6 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 3 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 4 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 3 from rod B to rod A.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 5 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 4 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 3 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 7 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 3 from rod B to rod A.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 4 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 5 from rod B to rod A.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 3 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 4 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 3 from rod B to rod A.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 6 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 3 from rod A to rod C.

Move disk 1 from rod B to rod A.

Move disk 2 from rod B to rod C.

Move disk 1 from rod A to rod C.

Move disk 4 from rod A to rod B.

Move disk 1 from rod C to rod B.

Move disk 2 from rod C to rod A.

Move disk 1 from rod B to rod A.

Move disk 3 from rod C to rod B.

Move disk 1 from rod A to rod C.

Move disk 2 from rod A to rod B.

Move disk 1 from rod C to rod B.
Move disk 5 from rod A to rod C.
Move disk 1 from rod B to rod A.
Move disk 2 from rod B to rod C.
Move disk 1 from rod A to rod C.
Move disk 3 from rod B to rod A.
Move disk 1 from rod C to rod B.
Move disk 2 from rod C to rod A.
Move disk 1 from rod B to rod A.
Move disk 4 from rod B to rod C.
Move disk 1 from rod A to rod C.
Move disk 2 from rod A to rod B.
Move disk 1 from rod C to rod B.
Move disk 3 from rod A to rod C.
Move disk 1 from rod B to rod A.
Move disk 2 from rod B to rod C.
Move disk 1 from rod A to rod C.

Process finished with exit code 0

```
#2. Python Program to Display Fibonacci Sequence Using Recursion.  
  
def fibo(n):  
    if (n <= 1):  
        return n  
    else:  
        return(fibo(n-1) + fibo(n-2))  
  
n = int(input("Enter number of terms: "))
```

```
if (n <= 0):  
    print("Enter positive number: ")  
else:  
    print("Fibonacci series: ")  
    for i in range(n):  
        print(fibo(i), end=" ")
```

Output:

Enter number of terms: 7

Fibonacci series:

0 1 1 2 3 5 8

Process finished with exit code 0

#3. Python Program to Determine How Many Times a Given Letter Occurs in a String Recursively.

```
def countch(s, ch):  
    if not s:  
        return 0  
    elif (s[0] == ch):  
        return countch(s[1:], ch) + 1  
  
    else:  
        return countch(s[1:], ch)
```

```
s = input("Enter string: ")
ch = input("Enter character to count: ")
print("Count: ", countch(s,ch))
```

Output:

Enter string: I am pursuing computer science and engineering at Navrachana University

Enter character to count: i

Count: 6

Process finished with exit code 0

#4. A lambda function to find the bigger number in two given numbers.

```
number_1 = int(input("Enter number1: "))
number_2 = int(input("Enter number2: "))
large = lambda number_1,number_2 : number_1>number_2
if (large(number_1,number_2) == True):
    print("number_1 is greater than number_2")
else:
    print("number 2 is greater than number 1")
```

Output:

Enter number1: 18

Enter number2: 81

number_2 is greater than number_1

Process finished with exit code 0

```
#5 Python program using filter () to filter out odd numbers from a list.  
  
lst = [1,2,3,4,5,6,7,8,9,0]  
odd = list(filter(lambda x: x % 2 == 1, lst))  
print("Odd numbers: ", odd)
```

Output:

Odd numbers: [1, 3, 5, 7, 9]

Process finished with exit code 0

```
#6. Write a Python program to convert all the characters in uppercase and lowercase and eliminate duplicate letters from a given sequence. Use map() function.
```

```
list1 = ["a","b","c","d","e","X","Y","Z","A","B","b","c"]
def case(s):
    if str(s).isupper():
        return str(s).lower()
    elif str(s).islower():
        return str(s).upper()
result = list(map(case, list1))
print("Changed: ", result)
result = set(result)
print("Unique characters: ", result)
```

Output:

Changed: ['A', 'B', 'C', 'D', 'E', 'x', 'y', 'z', 'a', 'b', 'B', 'C']

Unique characters: {'D', 'y', 'z', 'B', 'x', 'A', 'E', 'b', 'C', 'a'}

Process finished with exit code 0

```
#7. Python program to find the products of elements of two different lists  
using lambda function.
```

```
list1 = [1, 2, 3, 4, 5]  
list2 = [6, 7, 8, 9, 10]  
print("List1: ", list1)  
print("List2: ", list2)  
result = list(map(lambda num1, num2: num1 * num2, list1, list2))  
print("Final list: ", result)
```

Output:

List1: [1, 2, 3, 4, 5]

List2: [6, 7, 8, 9, 10]

Final list: [6, 14, 24, 36, 50]

Process finished with exit code 0

```
#8. Write a Python program to create a lambda function that adds 15 to a
given number passed in as an argument, also create a lambda function that
multiplies argument x with argument y and print the result.
a = 5
b = 6
c = 7
print("Initial Values: ")
print("x = ", a)
print("y = ", b)
print("a = ", c)
add15 = lambda c: c + 15
print("Add15: ", add15(c))
result = lambda a,y: a * b
print("Multiplication of x and y: ", result(a,b))
```

Output:

Initial Values:

x = 5

y = 6

a = 7

Add15: 22

Multiplication of x and y: 30

Process finished with exit code 0

```
#9. Write a Python program to find intersection of two given arrays using Lambda.
```

```
arr1 = [1, 2, 3, 4, 5, 6]
arr2 = [5, 6, 7, 8, 9]
print("Array1: ", arr1)
print("Array2: ", arr2)
result = list(filter(lambda x: x in arr1, arr2))
print ("Intersection: ", result)
```

Output:

Array1: [1, 2, 3, 4, 5, 6]

Array2: [5, 6, 7, 8, 9]

Intersection: [5, 6]

Process finished with exit code 0

```
#10. Write a Python program that sum the length of the names of a given list of names after removing the names that starts with a lowercase letter. Use lambda function.
```

```
name = ["Ajey", "Ashish", "Bhuvan", "sharma", "Beniwal"]  
filtername=list(filter(lambda el: el[0].isupper() and el[1:].islower(),  
name))  
print("Length of names: ", len("".join(filtername)))
```

Output:

Length of names: 23

Process finished with exit code 0