



**SNS COLLEGE OF ENGINEERING**  
Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade  
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai



**DEPARTMENT OF COMPUTER SCIENCE AND DESIGN**



# **19IT103 – COMPUTATIONAL THINKING AND PYTHON PROGRAMMING**

❖ A readable, dynamic, pleasant, flexible, fast and powerful language

## Recap:

### 1. Find a minimum in a list :

- One way is to sort the list of elements in ascending order and get the first element as minimum.
- Another method is to compare each element with other.
  - As an initial step, first element of the list is considered as minimum element.
  - And in each iteration, each element in the list is compared with the minimum.
  - If the element in the list is less than the minimum then swap both elements else compare with the next element in the list.
  - These steps are continued until the end of the list and finally print the minimum.

## Recap:

### 2. insert a card in a list of sorted cards :

- Start with an empty left hand and cards face down on the table.
- Then remove one card at a time from the table and Insert it into the correct position in the left hand.
- To find a correct position for a card, we compare it with each of the cards already in the hand from left to right.
- Once the position is found, the cards from that position are moved to the next higher indexed position and in that order.
- New card is inserted at the current position.

## 1.9 Illustrative problems:

- Guess an integer number in a range
- Tower of Hanoi

•

## 1.9.1 Guess an integer number in a range :



**Guess the same number that the computer has guessed.  
The number will range from 1 to 100.**

## 1.9.1 Guess an integer number in a range :

### Problem Statement:

- Shankar and Vijay are playing a game of integers.
- Shankar chooses an integer of his choice in his mind.
- It can be any integer value within the range of 1 to 100.
- Now, Vijay had to find that integer through some guesses.



## 1.9.1 Guess an integer number in a range :

### Problem Statement:

Shankar can provides 3 hints to Vijay, each hint can be one of the below types:

- Type 1 : Guess is Low
  - Type 2 : Guess is High
  - Type 3 : You guessed my number!
- 
- Now Vijay has to make some guesses in order to find Shankar's integer.
  - Note: Vijay is given only 10 chances to guess the number. If Vijay wins the game, then return **True** otherwise return **False**.

## 1.9.1 Guess an integer number in a range :

### Python coding:

```
import random

def ask_for_guess():
    '''returns an integer number as guessed by the user'''
    while True:
        guess = input('> ') # Enter the guess.

        if guess.isdecimal():
            return int(guess) # Convert string guess to an integer.
        print('Please enter a number between 1 and 100.')

print('*** Guess the Number ***')
print()
secretNumber = random.randint(1, 100) # Select a random number.
print('I am thinking of a number between 1 and 100.')
```



## 1.9.1 Guess an integer number in a range :

### Python coding:

```
for i in range(10): # Give the player 10 guesses.
    print('You have {} guesses left. Take a guess.'.format(10 - i))

    guess = ask_for_guess()
    if guess == secretNumber:
        break # Break out of the for loop if the guess is correct.

    # Offer a hint:
    if guess < secretNumber:
        print('Your guess is too low.')
    if guess > secretNumber:
        print('Your guess is too high.')

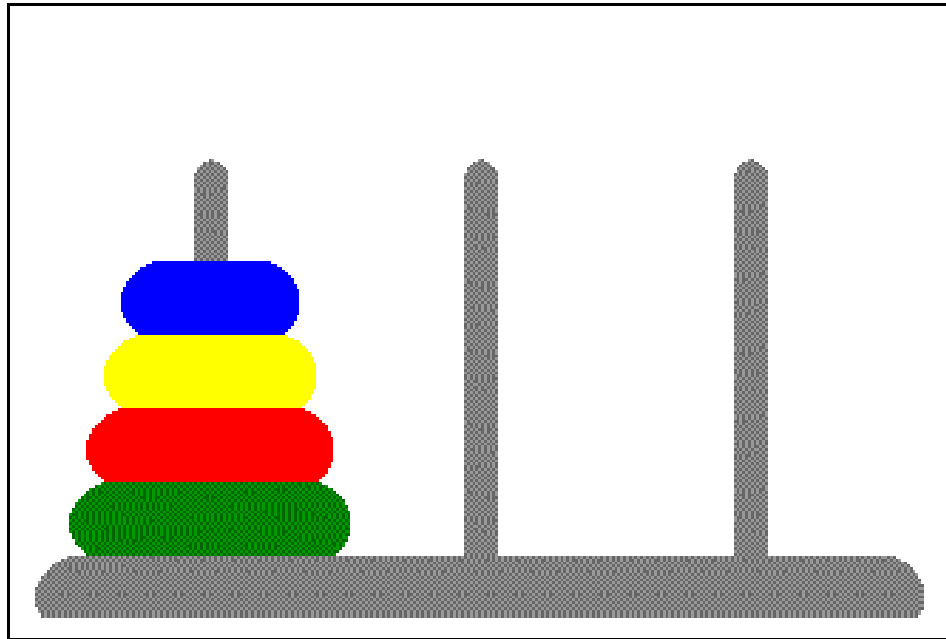
# Reveal the results:
if guess == secretNumber:
    print('Yay! You guessed my number!')
else:
    print('Game over. The number I was thinking of was', secretNumber)
```

## 1.9.1 Guess an integer number in a range :

### Output:

```
*** Guess the Number ***  
  
I am thinking of a number between 1 and 100.  
You have 10 guesses left. Take a guess.  
> 50  
Your guess is too high.  
You have 9 guesses left. Take a guess.  
> 25  
Your guess is too low.  
You have 8 guesses left. Take a guess.  
> 35  
Your guess is too low.  
You have 7 guesses left. Take a guess.  
> 45  
Your guess is too low.  
You have 6 guesses left. Take a guess.  
> 48  
Your guess is too low.  
You have 5 guesses left. Take a guess.  
> 49  
Yay! You guessed my number!
```

## 1.9.2 Tower of Hanoi :



## 1.9.2 Tower of Hanoi :

- The mission is to move all the disks to some another tower without violating the sequence of arrangement.
- A few rules to be followed for Tower of Hanoi are :
  - Only one disk can be moved among the towers at any given time.
  - Only the "top" disk can be removed.
  - No large disk can sit over a small disk.
- Tower of Hanoi puzzle with n disks can be solved in minimum  $2^n - 1$  steps.

## 1.9.2 Tower of Hanoi :

### Python Code:

```
# Tower of Hanoi

# one disk is tower A, destination is tower B, intermediate is tower C
print("Tower of Hanoi - with one disk")
source = 'A'
destination = 'B'
print("Move top disk from ", source, " to ", destination)
print()

# Two disk is at tower A, destination is tower B, intermediate in tower C
print("Tower of Hanoi - with 2 disk")
source = 'A'
destination = 'B'
temp = 'C'
print("Move top disk from ", source, " to ", temp)
print("Move top disk from ", source, " to ", destination)
print("Move top disk from ", temp, " to ", destination)
print()
```

## 1.9.2 Tower of Hanoi :

Output:

```
Tower of Hanoi - with one disk  
Move top disk from A to B  
  
Tower of Hanoi - with 2 disk  
Move top disk from A to C  
Move top disk from A to B  
Move top disk from C to B
```

## 1.9.2 Tower of Hanoi :

### Python Code:

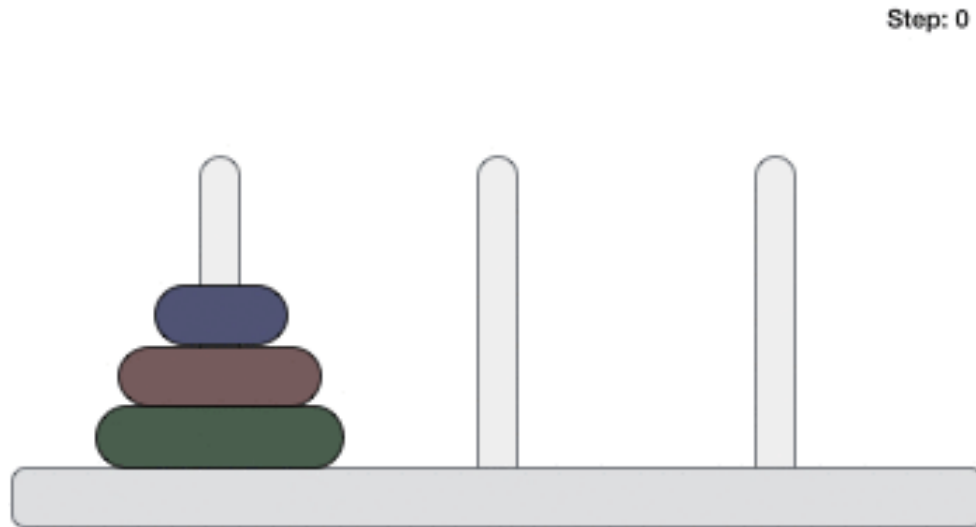
```
# In a recursive way

def tower_of_hanoi(n, fromTower, toTower, tempTower):
    if n == 1:
        print("Move top disc from ", fromTower, " to ", toTower)
    else:
        # Move n-1 disks from source to temp
        tower_of_hanoi(n - 1, fromTower, tempTower, toTower)
        # Move top disk from source to destination
        print("Move top disc from ", fromTower, " to ", toTower)
        # Move n-1 disks from temp to the destination
        tower_of_hanoi(n - 1, tempTower, toTower, fromTower)

n = int(input("Enter number of disks:"))
tower_of_hanoi(n, 'A', 'B', 'C')
```

## 1.9.2 Tower of Hanoi :

Output:





## 1.9.2 Tower of Hanoi :

Output:

```
Enter number of disks:3
Move top disc from A to B
Move top disc from A to C
Move top disc from B to C
Move top disc from A to B
Move top disc from C to A
Move top disc from C to B
Move top disc from A to B
```

## Summary:

### 1. Guess an integer number in a range :

- Shankar and Vijay are playing a game of integers.
- Shankar chooses an integer of his choice in his mind. It can be any integer value within the range of 1 to 100.
- Now, Vijay had to find that integer through some guesses.
- Shankar can provides 3 hints to Vijay, each hint can be one of the below types:
  - Type 1 : Guess is Low
  - Type 2 : Guess is High
  - Type 3 : You guessed my number!
- Now Vijay has to make some guesses in order to find Shankar's integer.
- *Note*: Vijay is given only 10 chances to guess the number. If Vijay wins the game, then return **True** otherwise return **False**.

## Summary:

### 2. Tower of Hanoi :

- Move all the disks to some another tower without violating the sequence of arrangement.
- Rules to be followed for Tower of Hanoi are :
  - Only one disk can be moved among the towers at any given time.
  - Only the "top" disk can be removed.
  - No large disk can sit over a small disk.
- Tower of Hanoi puzzle with n disks can be solved in **minimum  $2^n - 1$  steps.**

**THANK YOU**