**Practical No.4**

**Title :**

A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

**Objective :**

* To construct a tree and print the nodes.
* To find the time and space requirements of your method.

**Source Code :**

class TreeNode:

def \_\_init\_\_(self, name):

self.name = name

self.children = []

def build\_tree():

root\_name = input("Enter the name of the book: ")

book = TreeNode(root\_name)

while True:

chapter\_name = input("Enter the name of a chapter (or type 'exit' to finish): ")

if chapter\_name.lower() == 'exit':

break

chapter = TreeNode(chapter\_name)

book.children.append(chapter)

while True:

section\_name = input("Enter the name of a section (or type 'exit' to finish): ")

if section\_name.lower() == 'exit':

break

section = TreeNode(section\_name)

chapter.children.append(section)

while True:

subsection\_name = input("Enter the name of a subsection (or type 'exit' to finish): ")

if subsection\_name.lower() == 'exit':

break

subsection = TreeNode(subsection\_name)

section.children.append(subsection)

return book

def print\_tree(node, level=0):

print(" " \* level + node.name)

for child in node.children:

print\_tree(child, level + 1)

book = build\_tree()

# Print the tree

print("\nBook Structure:")

print\_tree(book)

**Output :**



