PRACTICAL

ARTIFICIAL • INTELLIGENCE (A.I)

SHEKHAR SUMAN

CERTIFICATE

Class: **T.Y.B.Sc.I.T.** Year: **2025 - 2026**

This is to certify that work entered in this Journal is the work of

Shri / ~~Kumari~~: **SHEKHAR SUMAN** .

of **T.Y.B.Sc.I.T** division _____ Roll No. **47** .

Uni. Exam No. _____ . has satisfactorily completed the

Required number of practical and worked for the **5TH** Semester

Of the year **2025 - 2026** in the college laboratory as laid down

By the university.

Head of the Department	External Examiner	Internal Examiner Subject teacher

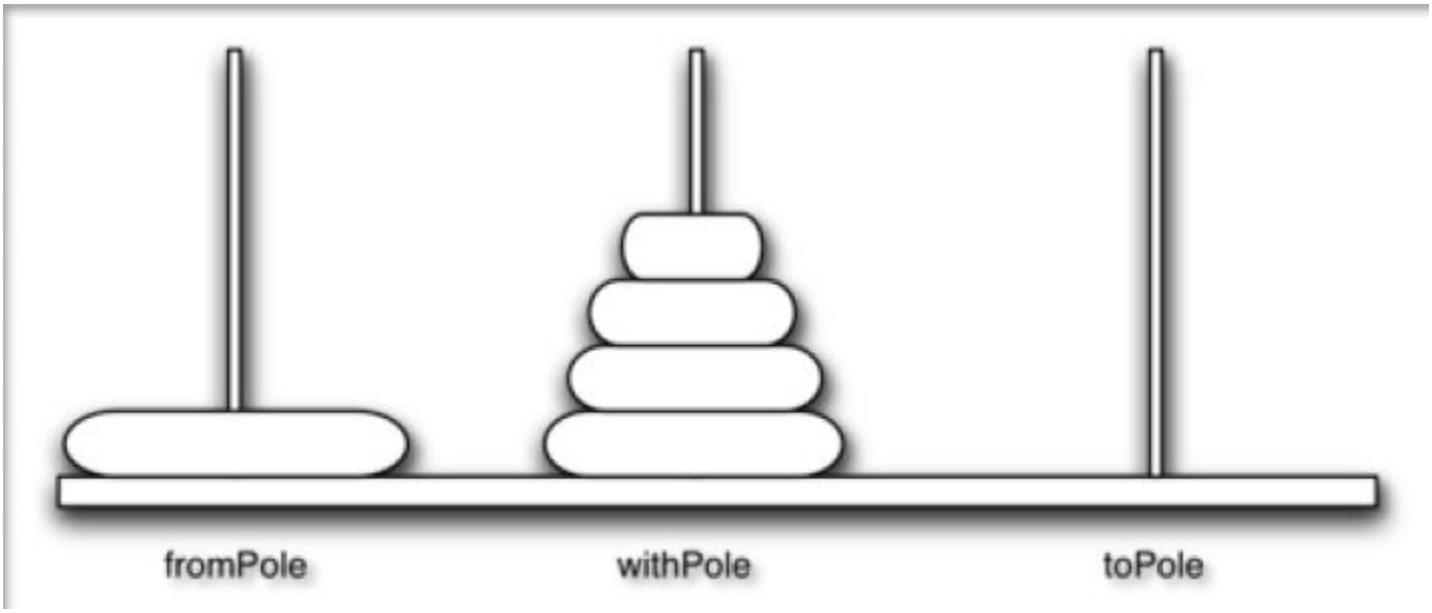
Date:

NAME: SHEKHAR SUMAN		INDEX
CLASS: T.Y.B.Sc.I.T.	**ROLL No**: 47	
SUBJECT: ARITIFICIAL INTELLIGENCE PRACTICAL		
COLLEGE: ST. ROCK'S DEGREE COLLEGE		

PR No:	TITLE	DATE	SIGN

PRACTICAL – 2B

AIM: - Write a program to solve tower of Hanoi problem.



CODE

```
print("Name: Shekhar Suman")
print("Roll No: 47\n")

def moveTower(height, fromPole, toPole, withPole):
    if height >= 1:
        moveTower(height - 1, fromPole, withPole, toPole)
        moveDisk(fromPole, toPole)
        moveTower(height - 1, withPole, toPole, fromPole)

def moveDisk(fp, tp):
    print("moving disk from", fp, "to", tp)

moveTower(3, "A", "B", "C")
```

OUTPUT

```
========== RESTART: E:\COLLEGE\TYBSCIT\A.I\hanoi.py ==========
Name: Shekhar Suman
Roll No: 47

moving disk from A to B
moving disk from A to C
moving disk from B to C
moving disk from A to B
moving disk from C to A
moving disk from C to B
moving disk from A to B
```

PRACTICAL – 3A

AIM: - Write a program to implement Alpha beta search.

```
tree = [[[5, 1, 2], [8, -8, -9]], [[9, 4, 5], [-3, 4, 3]]]
root = 0
pruned = 0

def children(branch, depth, alpha, beta):
    global tree, root, pruned
    i = 0
    for child in branch:
        if isinstance(child, list):
            nalpha, nbeta = children(child, depth + 1, alpha, beta)
            if depth % 2 == 1:
                beta = min(beta, nalpha)
            else:
                alpha = max(alpha, nbeta)
            branch[i] = alpha if depth % 2 == 0 else beta
        else:
            if depth % 2 == 0 and alpha < child:
                alpha = child
            if depth % 2 == 1 and beta > child:
                beta = child
        if alpha >= beta:
            pruned += 1
            break
        i += 1

    if depth == root:
        tree = alpha if root == 0 else beta
    return (alpha, beta)

def alphabeta(in_tree=None, start=root, upper=-15, lower=15):
    global tree, pruned, root
    if in_tree is not None:
        tree = in_tree
    alpha, beta = children(tree, start, upper, lower)

    if __name__ == "__main__":
        print("Name: Shekhar Suman")
        print("Roll No: 47")
        print("(alpha, beta):", alpha, beta)
        print("Result:", tree)
        print("Times pruned:", pruned)
    return (alpha, beta, tree, pruned)

if __name__ == "__main__":
    alphabeta()
```

OUTPUT

```
================ RESTART: E:\COLLEGE\TYBSCIT\A.I\PRACTICAL_3A.py ===============
Name: Shekhar Suman
Roll No: 47
(alpha, beta): 5 15
Result: 5
Times pruned: 2
>
```

PRACTICAL – 7A

AIM: - Write a program to implement Alpha beta search.

```
import random

suits = ["Hearts", "Diamonds", "Clubs", "Spades"]
royals = ["J", "Q", "K", "A"]

cardfaces = [str(i) for i in range(2, 11)] + royals

deck = []
for suit in suits:
    for face in cardfaces:
        deck.append(face + " of " + suit)

random.shuffle(deck)

print("Name: Shekhar Suman")
print("Roll No: 47\n")

for card in deck:
    print(card)

import itertools, random

deck = list(itertools.product(
    ['2','3','4','5','6','7','8','9','10','J','Q','K','A'],
    ['Spade','Heart','Diamond','Club']
))

random.shuffle(deck)

print("\nYou got:")
for i in range(5):
    print(deck[i][0], "of", deck[i][1])
```

OUTPUT

```
PS C:\Users\SHEKHAR SUMAN> python -u "e:\COLLEGE\TYBSCIT\A.I\PRACTICAL_7.py"
Name: Shekhar Suman
Roll No: 47

10 of Hearts
10 of Clubs
2 of Spades
K of Clubs
A of Diamonds
A of Clubs
10 of Diamonds
K of Spades
5 of Spades
7 of Spades
Q of Diamonds
8 of Diamonds
9 of Diamonds
Q of Hearts
4 of Diamonds
8 of Hearts
8 of Clubs
2 of Clubs
A of Hearts
J of Diamonds
3 of Hearts
5 of Hearts
J of Hearts
5 of Diamonds
7 of Diamonds
4 of Spades
9 of Clubs
4 of Hearts
A of Spades
6 of Clubs
7 of Clubs
6 of Hearts
K of Diamonds
J of Clubs
10 of Spades
2 of Hearts
9 of Hearts
7 of Hearts
9 of Spades
6 of Spades
K of Hearts
2 of Diamonds
5 of Clubs
3 of Clubs
Q of Clubs
J of Spades
6 of Diamonds
Q of Spades
8 of Spades
4 of Clubs
3 of Diamonds
3 of Spades

You got:
2 of Club
10 of Heart
K of Heart
A of Spade
Q of Spade
PS C:\Users\SHEKHAR SUMAN>
```