S.No	Experiments	Date of experiment	Date of Submission	Remarks
1.	1.Write a Program to Check Prime Number 2. Write a Program to Print the Fibonacci sequence. 3. Write a Program to Find the Factorial of a Number. 4. Write a program to reverse digits of a number. 5. Write Program in python to swap two numbers.			
2.	Write a program to implement the Tic-Tac-Toe problem.			
3.	Write a program to Implement a single Player game.			
4.	Write a program to implement a water jug problem in python.			
5.	Implement a Brute force solution to the Knapsack problem in Python.			
6.	Write a program to implement A* algorithm in python			
7.	Write a program to implement BFS for water jug problem using Python			
8.	Write a program to implement DFS using Python.			
9.	Design an XOR truth table using Python			
10.	Tokenization of word and Sentences with the help of NLTK package.			

Experiment 9

Aim: Design an XOR truth table using Python

Language Used: Python

Theory: The XOR (Exclusive OR) operation is a logical function that produces a true (1) output only when the input values differ. When both inputs are identical, the output is false (0).

Source Code:

```
def xor(a: int, b: int) -> int:
    return a ^ b
def xor_truth_table() -> None:
    print("XOR Truth Table")
    print("================")
    print("A | B | A XOR B")
    print("--|---|--------")
    for a in [0, 1]:
        for b in [0, 1]:
            result = xor(a, b)
            print(f"{a} | {b} |    {result}")
    print("\nExplanation:")
    print("- XOR (Exclusive OR) returns 1 if inputs are different, otherwise 0.")
    print("- When A and B are both 0 or both 1, XOR is 0.")
    print("- When A and B are different (0,1) or (1,0), XOR is 1.")
if __name__ == "__main__":
    xor_truth_table()
```

Output:

```
XOR Truth Table
================
A | B | A XOR B
--|---|--------
0 | 0 |    0
0 | 1 |    1
1 | 0 |    1
1 | 1 |    0

Explanation:
- XOR (Exclusive OR) returns 1 if inputs are different, otherwise 0.
- When A and B are both 0 or both 1, XOR is 0.
- When A and B are different (0,1) or (1,0), XOR is 1.
```

Experiment 10

Aim: Tokenization of word and Sentences with the help of NLTK package.

Language used: Python

Theory: Tokenization is the process of dividing text into smaller units like words or sentences. The Natural Language Toolkit (NLTK) offers built-in functions for tokenization:

1. **Sentence Tokenization**: Divides text into individual sentences.
2. **Word Tokenization**: Breaks sentences down into words.

Source Code:

```
import nltk

from nltk.tokenize import word_tokenize, sent_tokenize

nltk.download('punkt')

nltk.download('punkt_tab') # Download the punkt_tab data package

text = "Natural Language Processing (NLP) is a fascinating field. It enables machines to understand human language! Tokenization is an essential step in NLP."

# Sentence Tokenization: Splitting the text into individual sentences

sentences = sent_tokenize(text)

print("Sentence Tokenization:")

for i, sentence in enumerate(sentences, 1):

  print(f"{i}. {sentence}")

# Word Tokenization: Splitting each sentence into words

words = word_tokenize(text)

print("\nWord Tokenization:")

print(words)
```

Output: