17. Table of a Number

```
n = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{n} x {i} = {n * i}")
```

```
Enter a number: 2
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

18. Swap Two Numbers (without third variable)

```
a = float(input("Enter first number: "))  # Changed to float to handle potential decimal input
b = float(input("Enter second number: "))  # Changed to float to handle potential decimal input
a, b = b, a
print("Swapped values:", a, b)
```

```
Enter first number: 10
Enter second number: 20
Swapped values: 20.0 10.0
```

19. Check Substring

```
s1 = input("Enter main string: ")
s2 = input("Enter substring: ")
print(s2 in s1)
```

```
Enter main string: Main flow
Enter substring: flow
True
```

20. Decimal to Binary

```
n = int(input("Enter a decimal number: "))
print(bin(n)[2:])
```

```
Enter a decimal number: 10
1010
```

21. Matrix Addition

```
A = [[1, 2], [3, 4]]
B = [[5, 6], [7, 8]]
result = [[A[i][j] + B[i][j] for j in range(len(A[0]))] for i in range(len(A))]
print(result)
```

```
[[6, 8], [10, 12]]
```

22. Matrix Multiplication

```
A = [[1, 2], [3, 4]]
B = [[5, 6], [7, 8]]
result = [[sum(A[i][k] * B[k][j] for k in range(len(B))) for j in range(len(B[0]))] for i in range(len(A))]
print(result)
```

```
[[19, 22], [43, 50]]
```

23. Find Second Largest

```
nums = list(map(int, input("Enter numbers: ").split()))
nums = list(set(nums))  # Remove duplicates
nums.sort()
print(nums[-2] if len(nums) > 1 else "No second largest")
```

```
Enter numbers: 10 30 40 50
40
```

24. Check Anagram

```
s1 = input("Enter first string: ")
s2 = input("Enter second string: ")
print(sorted(s1) == sorted(s2))
```

```
Enter first string: listen
Enter second string: silent
True
```

3. AI-Based Tic-Tac-Toe

```
import random

board = [" "] * 9
player = "X"
computer = "O"

def print_board():
    for i in range(0, 9, 3):
        print(board[i], "|", board[i+1], "|", board[i+2])
    print()

def check_winner():
    win_patterns = [(0,1,2), (3,4,5), (6,7,8), (0,3,6), (1,4,7), (2,5,8), (0,4,8), (2,4,6)]
    for p in win_patterns:
        if board[p[0]] == board[p[1]] == board[p[2]] and board[p[0]] != " ":
            return board[p[0]]
    return None

def computer_move():
    empty = [i for i in range(9) if board[i] == " "]
    move = random.choice(empty)
    board[move] = computer

def play():
    print_board()
    while " " in board:
        move = int(input("Enter position (0-8): "))
        if board[move] == " ":
            board[move] = player
        else:
            print("Invalid move, try again.")
            continue
        print_board()
        if check_winner():
            print("Player Wins!")
            return
        if " " not in board:
            break
        computer_move()
        print_board()
        if check_winner():
            print("Computer Wins!")
            return
    print("It's a draw!")

play()
```

```
  |   |  
  |   |  
  |   |  

X |   |  
  |   |  
  |   |  

X |   |  
O |   |  
  |   |  
X | X | 
O |   | 
  |   | 

X | X | O
O |   | 
  |   | 

Invalid move, try again.
Invalid move, try again.
Enter position (0-8): 
```