Started on	Saturday, 26 April 2025, 3:16 PM
State	Finished
Completed on	Saturday, 26 April 2025, 3:22 PM
Time taken	5 mins 51 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program to implement pattern matching on the given string using Brute Force algorithm.

For example:

Test	Input	Result
BF(a1,a2)	abcaaaabbbbcccabcbabdbcsbbbbbnnn ccabcba	12

Answer: (penalty regime: 0 %)

Reset answer

```
def BF(s1,s2):
    n = len(s1)
    m = len(s2)

    for i in range(n - m + 1):
        j = 0
        while j < m and s1[i + j] == s2[j]:
            j += 1
        if j == m:
            return i
    return -1
if __name__ == "__main__":
    a1=input()
    a2=input()
    b=BF(a1,a2)
    print(b)
```

	Test	Input	Expected	Got	
✓	BF(a1,a2)	abcaaaabbbbcccabcbabdbcsbbbbbnnn ccabcba	12	12	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **2**

Correct

Mark 20.00 out of 20.00

Write a python program to implement knight tour problem using backtracking

For example:

Input	Result
5	Found a solution 01 20 11 14 03 10 15 02 19 12 21 24 13 04 07 16 09 06 23 18 25 22 17 08 05

Answer: (penalty regime: 0 %)

Reset answer

```python

BOARD_SIZE = int(input())
board = [[0 for i in range(BOARD_SIZE)] for j in range(BOARD_SIZE)]
STEPS = [[-1, 2], [1, 2], [-2, 1], [2, 1], [1, -2], [-1, -2], [2, -1], [-2, -1]]


def solve_knights_tour(x, y, step_count):
    if step_count > BOARD_SIZE * BOARD_SIZE:
        return True

    for step in STEPS:
        next_x = x + step[0]
        next_y = y + step[1]

        if is_safe(next_x, next_y):
            board[next_x][next_y] = step_count
            if solve_knights_tour(next_x, next_y, step_count + 1):
                return True
            board[next_x][next_y] = 0

    return False

```

	Input	Expected	Got	
✓	5	Found a solution 01 20 11 14 03 10 15 02 19 12 21 24 13 04 07 16 09 06 23 18 25 22 17 08 05	Found a solution 01 20 11 14 03 10 15 02 19 12 21 24 13 04 07 16 09 06 23 18 25 22 17 08 05	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

Not answered

Mark 0.00 out of 20.00

Write a Program for Implementing merge sort on float values using python recursion.

For example:

Test	Input	Result
`merge_sort(inp_arr)`	5 3.2 1.6 9.5 4.3 4.55	Input Array: [3.2, 1.6, 9.5, 4.3, 4.55] Sorted Array: [1.6, 3.2, 4.3, 4.55, 9.5]
`merge_sort(inp_arr)`	6 3.2 1.2 5.3 9.6 8.5 7.4	Input Array: [3.2, 1.2, 5.3, 9.6, 8.5, 7.4] Sorted Array: [1.2, 3.2, 5.3, 7.4, 8.5, 9.6]

Answer: (penalty regime: 0 %)

Question **4**

Correct

Mark 20.00 out of 20.00

Write a python program to implement KMP (Knuth Morris Pratt).

For example:

Input	Result
ABABDABACDABABCABAB ABABCABAB	Found pattern at index 10

Answer: (penalty regime: 0 %)

Reset answer

```python

def KMPSearch(pat, txt):
    M = len(pat)
    N = len(txt)

    lps = [0] * M
    computeLPSArray(pat, M, lps)

    i = 0
    j = 0
    while i < N:
        if pat[j] == txt[i]:
            i += 1
            j += 1

        if j == M:
            print("Found pattern at index", i - j)
            j = lps[j - 1]

        elif i < N and pat[j] != txt[i]:
            if j != 0:
                j = lps[j - 1]
```

	Input	Expected	Got	
✓	ABABDABACDABABCABAB ABABCABAB	Found pattern at index 10	Found pattern at index 10	✓
✓	SAVEETHAENGINEERING VEETHA	Found pattern at index 2	Found pattern at index 2	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **5**

Correct

Mark 20.00 out of 20.00

Write a python program to check whether Hamiltonian path exits in the given graph.

For example:

Test	Result
Hamiltonian_path(adj, N)	YES

Answer: (penalty regime: 0 %)

Reset answer

```

def Hamiltonian_path(adj, N):
    dp = [[False for _ in range(1 << N)] for _ in range(N)]

    for i in range(N):
        dp[i][1 << i] = True

    for i in range(1 << N):
        for j in range(N):
            if (i & (1 << j)) and any((i & (1 << k)) and adj[k][j] and j != k and dp[k][i ^ (1 << j)]
                dp[j][i] = True

    return any(dp[i][(1 << N) - 1] for i in range(N))

adj = [ [ 0, 1, 1, 1, 0 ] ,
        [ 1, 0, 1, 0, 1 ],
        [ 1, 1, 0, 1, 1 ],
        [ 1, 0, 1, 0, 0 ] ]

N = len(adj)

```

	Test	Expected	Got	
✓	Hamiltonian_path(adj, N)	YES	YES	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.