

<b>Started on</b>	Friday, 11 April 2025, 1:16 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 11 April 2025, 2:59 PM
<b>Time taken</b>	1 hour 42 mins
<b>Grade</b>	<b>80.00</b> out of 100.00

### Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement Boyer Moore Algorithm with Good Suffix heuristic to find pattern in given text string.

For example:

Input	Result
ABAAABAACD	pattern occurs at shift = 0
ABA	pattern occurs at shift = 4

Answer: (penalty regime: 0 %)

Reset answer

```

1 def preprocess_strong_suffix(shift, bpos, pat, m):
2
3     i = m
4     j = m + 1
5     bpos[i] = j
6     while i > 0:
7         while j <= m and pat[i - 1] != pat[j - 1]:
8             if shift[j] == 0:
9                 shift[j] = j - i
10                j = bpos[j]
11            i -= 1
12            j -= 1
13        bpos[i] = j
14
15 def preprocess_case2(shift, bpos, pat, m):
16     j = bpos[0]
17     for i in range(m + 1):
18         if shift[i] == 0:
19             shift[i] = j
20         if i == j:
21             j = bpos[j]
22 def search(text, pat):

```

	Input	Expected	Got	
✓	ABAAABAACD ABA	pattern occurs at shift = 0 pattern occurs at shift = 4	pattern occurs at shift = 0 pattern occurs at shift = 4	✓
✓	SaveethaEngineering veetha	pattern occurs at shift = 2 pattern occurs at shift = 22	pattern occurs at shift = 2 pattern occurs at shift = 22	✓

Passed all tests! ✓

Comment

Marks for this submission: 20.00/20.00.

## Question 2

Correct

Mark 20.00 out of 20.00

Create a python program to find the Hamiltonian path using Depth First Search for traversing the graph .

For example:

Test	Result
hamiltonian.findCycle()	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']

Answer: (penalty regime: 0 %)

Reset answer

```

1 class Hamiltonian:
2     def __init__(self, start):
3         self.start = start
4         self.cycle = []
5         self.hasCycle = False
6
7     def findCycle(self):
8         self.cycle.append(self.start)
9         self.solve(self.start)
10
11    def solve(self, vertex):
12        if vertex == self.start and len(self.cycle) == N+1:
13            self.hasCycle = True
14            self.displayCycle()
15            return
16        for i in range(len(vertices)):
17            if adjacencyM[vertex][i] == 1 and visited[i] == 0:
18                nbr = i
19                visited[nbr] = 1
20                self.cycle.append(nbr)
21                self.solve(nbr)
22                visited[nbr] = 0

```

	Test	Expected	Got	
✓	hamiltonian.findCycle()	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']	['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'A'] ['A', 'H', 'G', 'F', 'E', 'D', 'C', 'B', 'A']	✓

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 python program to implement merge sort without using recursive function on the given list of float values.

**For example:**

Input	Result
5	left: [6.2]
6.2	Right: [4.1]
4.1	left: [3.2]
3.2	Right: [5.6]
5.6	left: [7.4]
7.4	Right: []
	left: [4.1, 6.2]
	Right: [3.2, 5.6]
	left: [7.4]
	Right: []
	left: [3.2, 4.1, 5.6, 6.2]
	Right: [7.4]
	[3.2, 4.1, 5.6, 6.2, 7.4]
6	left: [3.2]
3.2	Right: [8.9]
8.9	left: [4.5]
4.5	Right: [6.2]
6.2	left: [1.5]
1.5	Right: [8.0]
8.0	left: [3.2, 8.9]
	Right: [4.5, 6.2]
	left: [1.5, 8.0]
	Right: []
	left: [3.2, 4.5, 6.2, 8.9]
	Right: [1.5, 8.0]
	[1.5, 3.2, 4.5, 6.2, 8.0, 8.9]

**Answer:** (penalty regime: 0 %)

1	
---	--

## Question 4

Correct

Mark 20.00 out of 20.00

Write a python program to implement knight tour problem using warnsdorff's algorithm

For example:

Test	Input	Result
a.warnsdorff((x,y))	8 8 3 3	board: [21, 32, 17, 30, 39, 36, 15, 42] [18, 29, 20, 35, 16, 41, 54, 37] [33, 22, 31, 40, 53, 38, 43, 14] [28, 19, 34, 1, 44, 49, 60, 55] [23, 2, 27, 52, 61, 56, 13, 50] [8, 5, 24, 45, 48, 51, 62, 59] [3, 26, 7, 10, 57, 64, 47, 12] [6, 9, 4, 25, 46, 11, 58, 63]

Answer: (penalty regime: 0 %)

Reset answer

```

1 KNIGHT_MOVES = [(2, 1), (1, 2), (-1, 2), (-2, 1), (-2, -1), (-1, -2), (1, -2), (2, -1)]
2 class KnightTour:
3     def __init__(self, board_size):
4         self.board_size = board_size
5         self.board = []
6         for i in range(board_size[0]):
7             temp = []
8             for j in range(board_size[1]):
9                 temp.append(0)
10            self.board.append(temp)
11            self.move = 1
12
13        def print_board(self):
14            print('board:')
15            for i in range(self.board_size[0]):
16                print(self.board[i])
17
18        def warnsdorff(self, start_pos, GUI=False):
19
20            x_pos, y_pos = start_pos
21            self.board[x_pos][y_pos] = self.move
22

```

	Test	Input	Expected	Got	
✓	a.warnsdorff((x,y))	8 8 3 3	board: [21, 32, 17, 30, 39, 36, 15, 42] [18, 29, 20, 35, 16, 41, 54, 37] [33, 22, 31, 40, 53, 38, 43, 14] [28, 19, 34, 1, 44, 49, 60, 55] [23, 2, 27, 52, 61, 56, 13, 50] [8, 5, 24, 45, 48, 51, 62, 59] [3, 26, 7, 10, 57, 64, 47, 12] [6, 9, 4, 25, 46, 11, 58, 63]	board: [21, 32, 17, 30, 39, 36, 15, 42] [18, 29, 20, 35, 16, 41, 54, 37] [33, 22, 31, 40, 53, 38, 43, 14] [28, 19, 34, 1, 44, 49, 60, 55] [23, 2, 27, 52, 61, 56, 13, 50] [8, 5, 24, 45, 48, 51, 62, 59] [3, 26, 7, 10, 57, 64, 47, 12] [6, 9, 4, 25, 46, 11, 58, 63]	✓

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 implement pattern matching on the given string using Brute Force algorithm.

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```
1 def BF(s1,s2):
2
3     i = 0
4     j = 0
5     while(i < len(s1) and j < len(s2)):
6         if(s1[i] == s2[j]):
7             i += 1
8             j += 1
9         else:
10            i = i - j + 1
11            j = 0
12     if(j >= len(s2)):
13         return i - len(s2)
14     else:
15         return 0
16
17 if __name__ == "__main__":
18     a1=input()
19     a2=input()
20     b=BF(a1,a2)
21     print(b)
22 ##
```

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

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.