
Started on Wednesday, 14 May 2025, 11:23 AM

State Finished

Completed on Wednesday, 14 May 2025, 11:53 AM

Time taken 30 mins 22 secs

Grade 100.00 out of 100.00

Question 1

Correct

Mark 20.00 out of 20.00

Write a python program to implement knight tour problem

For example:

Input	Result
5	[1, 12, 25, 18, 3]
5	[22, 17, 2, 13, 24]
	[11, 8, 23, 4, 19]
	[16, 21, 6, 9, 14]
	[7, 10, 15, 20, 5]
	[(0, 0), (1, 2), (0, 4), (2, 3), (4, 4), (3, 2), (4, 0), (2, 1), (3, 3), (4, 1), (2, 0), (0, 1), (1, 3), (3, 4), (4, 2), (3, 0), (1, 1), (0, 3), (2, 4), (4, 3), (3, 1), (1, 0), (2, 2), (1, 4), (0, 2)]
	Done!

Answer: (penalty regime: 0 %)

Reset answer

```

1 import sys
2
3 class KnightsTour:
4     def __init__(self, width, height):
5         self.w = width
6         self.h = height
7         self.board = []
8         self.generate_board()
9
10    def generate_board(self):
11        for i in range(self.h):
12            self.board.append([0]*self.w)
13
14    def print_board(self):
15        for elem in self.board:
16            print(elem)
17
18    def generate_legal_moves(self, cur_pos):
19        possible_pos = []
20        move_offsets = [(1, 2), (1, -2), (-1, 2), (-1, -2),
21                        (2, 1), (2, -1), (-2, 1), (-2, -1)]
22

```

	Input	Expected	Got	
✓	5 5	[1, 12, 25, 18, 3] [22, 17, 2, 13, 24] [11, 8, 23, 4, 19] [16, 21, 6, 9, 14] [7, 10, 15, 20, 5] [(0, 0), (1, 2), (0, 4), (2, 3), (4, 4), (3, 2), (4, 0), (2, 1), (3, 3), (4, 1), (2, 0), (0, 1), (1, 3), (3, 4), (4, 2), (3, 0), (1, 1), (0, 3), (2, 4), (4, 3), (3, 1), (1, 0), (2, 2), (1, 4), (0, 2)] Done!	[1, 12, 25, 18, 3] [22, 17, 2, 13, 24] [11, 8, 23, 4, 19] [16, 21, 6, 9, 14] [7, 10, 15, 20, 5] [(0, 0), (1, 2), (0, 4), (2, 3), (4, 4), (3, 2), (4, 0), (2, 1), (3, 3), (4, 1), (2, 0), (0, 1), (1, 3), (3, 4), (4, 2), (3, 0), (1, 1), (0, 3), (2, 4), (4, 3), (3, 1), (1, 0), (2, 2), (1, 4), (0, 2)] Done!	✓

	Input	Expected	Got	
✓	6 6	[1, 32, 9, 18, 3, 34] [10, 19, 2, 33, 26, 17] [31, 8, 25, 16, 35, 4] [20, 11, 36, 27, 24, 15] [7, 30, 13, 22, 5, 28] [12, 21, 6, 29, 14, 23] [(0, 0), (1, 2), (0, 4), (2, 5), (4, 4), (5, 2), (4, 0), (2, 1), (0, 2), (1, 0), (3, 1), (5, 0), (4, 2), (5, 4), (3, 5), (2, 3), (1, 5), (0, 3), (1, 1), (3, 0), (5, 1), (4, 3), (5, 5), (3, 4), (2, 2), (1, 4), (3, 3), (4, 5), (5, 3), (4, 1), (2, 0), (0, 1), (1, 3), (0, 5), (2, 4), (3, 2)] Done!	[1, 32, 9, 18, 3, 34] [10, 19, 2, 33, 26, 17] [31, 8, 25, 16, 35, 4] [20, 11, 36, 27, 24, 15] [7, 30, 13, 22, 5, 28] [12, 21, 6, 29, 14, 23] [(0, 0), (1, 2), (0, 4), (2, 5), (4, 4), (5, 2), (4, 0), (2, 1), (0, 2), (1, 0), (3, 1), (5, 0), (4, 2), (5, 4), (3, 5), (2, 3), (1, 5), (0, 3), (1, 1), (3, 0), (5, 1), (4, 3), (5, 5), (3, 4), (2, 2), (1, 4), (3, 3), (4, 5), (5, 3), (4, 1), (2, 0), (0, 1), (1, 3), (0, 5), (2, 4), (3, 2)] Done!	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 2

Correct

Mark 20.00 out of 20.00

Create a python program to implement Hamiltonian circuit problem using Backtracking.

For example:

Result

Solution Exists: Following is one Hamiltonian Cycle
0 1 2 4 3 0

Answer: (penalty regime: 0 %)

Reset answer

```

1 class Graph():
2     def __init__(self, vertices):
3         self.graph = [[0 for column in range(vertices)]
4                       for row in range(vertices)]
5         self.V = vertices
6
7     def isSafe(self, v, pos, path):
8         if self.graph[path[pos-1]][v] == 0:
9             return False
10        for vertex in path:
11            if vertex == v:
12                return False
13
14        return True
15
16    def hamCycleUtil(self, path, pos):
17        if pos==self.V:
18            return True
19        for v in range(1,self.V):
20            if self.isSafe(v,pos,path):
21                path[pos]=v
22                if self.hamCycleUtil(path,pos+1):

```

	Expected	Got	
✓	Solution Exists: Following is one Hamiltonian Cycle 0 1 2 4 3 0	Solution Exists: Following is one Hamiltonian Cycle 0 1 2 4 3 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question 3

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

```

1 def KMPSearch(pat, txt):
2     lp=len(pat)
3     ls=len(txt)
4     lps=[0]*lp
5     computeLPSArray(pat,lp,lps)
6     i=0
7     j=0
8
9     while(i!=ls):
10        if txt[i]==pat[j]:
11            i+=1
12            j+=1
13        else:
14            j=lps[j-1]
15        if j==lp:
16            print("Found pattern at index",i-j)
17            j=lps[j-1]
18        elif j==0:
19            i+=1
20
21 def computeLPSArray(pat, M, lps):
22     len = 0

```

	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 4

Correct

Mark 20.00 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 def merge_sort(arr):
2     stack = [[val] for val in arr]
3
4     while len(stack) > 1:
5         temp_stack = []
6         for i in range(0, len(stack), 2):
7             left = stack[i]
8             right = stack[i + 1] if i + 1 < len(stack) else []
9             merged = merge(left, right)
10            temp_stack.append(merged)
11            print(f"left: {left}")
12            print(f"Right: {right}")
13            stack = temp_stack
14
15        return stack[0]
16
17 def merge(left, right):
18     i = j = 0
19     li = []
20
21     while i < len(left) and j < len(right):
22         if left[i] < right[j]:

```

	Input	Expected	Got	
✓	5 6.2 4.1 3.2 5.6 7.4	left: [6.2] Right: [4.1] left: [3.2] Right: [5.6] left: [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]	left: [6.2] Right: [4.1] left: [3.2] Right: [5.6] left: [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 3.2 8.9 4.5 6.2 1.5 8.0	left: [3.2] Right: [8.9] left: [4.5] Right: [6.2] left: [1.5] Right: [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]	left: [3.2] Right: [8.9] left: [4.5] Right: [6.2] left: [1.5] Right: [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]	✓

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)	abcaaaabbbbcccabcbabdbcsbbbbnnn ccabcba	12

Answer: (penalty regime: 0 %)

Reset answer

```

1 def BF(s1,s2):
2     for i in range(len(s1)-len(s2)):
3         split = len(s2)
4         if s1[i:(i+split)] == s2:
5             return i
6
7 if __name__ == "__main__":
8     a1=input()
9     a2=input()
10    b=BF(a1,a2)
11    print(b)
12

```

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

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.