Started on	Thursday, 10 April 2025, 2:13 PM
State	Finished
Completed on	Thursday, 10 April 2025, 4:18 PM
Time taken	2 hours 5 mins
Overdue	5 mins 3 secs
Grade	**100.00** out of 100.00

Question **1**

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

```
class Hamiltonian:
    def __init__(self, start):
        self.start = start
        self.cycle = []
        self.hasCycle = False

    def findCycle(self):
        self.cycle.append(self.start)
        self.solve(self.start)

    def solve(self, vertex):
        ############  Add your code here ###############
        if vertex==self.start and len(self.cycle)==N+1:
            self.hasCycle=True
            self.displayCycle()
        for i in range(len(vertices)):
            if adjacencyM[vertex][i]==1 and visited[i]==0:
                nbr=i
                self.cycle.append(nbr)
                visited[nbr]=1
                self.solve(nbr)
                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 **2**

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.warnsdroff((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

```
KNIGHT_MOVES = [(2, 1), (1, 2), (-1, 2), (-2, 1), (-2, -1), (-1, -2), (1, -2), (2, -1)]
class KnightTour:
    def __init__(self, board_size):
        self.board_size = board_size  # tuple
        self.board = []
        for i in range(board_size[0]):
            temp = []
            for j in range(board_size[1]):
                temp.append(0)
            self.board.append(temp) # empty cell
        self.move = 1

    def print_board(self):
        print('board:')
        for i in range(self.board_size[0]):
            print(self.board[i])

    def warnsdroff(self, start_pos, GUI=False):
    ######################### Add your code here ############################3
        x_pos,  y_pos = start_pos
        self.board[x_pos][y_pos] = self.move

```

	Test	Input	Expected	Got	
✔	a.warnsdroff((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 **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

```
def KMPSearch(pat, txt):
    ############## Add your code here #################
    lp=len(pat)
    ls=len(txt)
    lps=[0]*lp
    computeLPSArray(pat,lp,lps)
    i=0
    j=0

    while(i!=ls):
        if txt[i]==pat[j]:
            i+=1
            j+=1
        else:
            j=lps[j-1]
        if j==lp:
            print("Found pattern at index",i-j)
            j=lps[j-1]
        elif j==0:
            i+=1

def computeLPSArray(pat, M, lps):
```

	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 quick sort on the given float values and print the sorted list and pivot value of each iteration.

For example:

Input	Result
5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]
4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]

Answer: (penalty regime: 0 %)

```
def quicksort(arr, low, high):
    if low < high:
        pi = partition(arr, low, high)
        quicksort(arr, low, pi - 1)
        quicksort(arr, pi + 1, high)

def partition(arr, low, high):
    pivot = arr[low]
    left = low + 1
    right = high

    while True:
        while left <= right and arr[left] <= pivot:
            left = left + 1
        while left <= right and arr[right] >= pivot:
            right = right - 1
        if left <= right:
            arr[left], arr[right] = arr[right], arr[left]
        else:
            break

    arr[low], arr[right] = arr[right], arr[low]
```

	Input	Expected	Got	
✔	5 2.3 3.2 1.6 4.2 3.9	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	Input List [2.3, 3.2, 1.6, 4.2, 3.9] pivot: 2.3 pivot: 3.2 pivot: 4.2 Sorted List [1.6, 2.3, 3.2, 3.9, 4.2]	✔
✔	4 5 2 49 3	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	Input List [5.0, 2.0, 49.0, 3.0] pivot: 5.0 pivot: 3.0 Sorted List [2.0, 3.0, 5.0, 49.0]	✔

	Input	Expected	Got	
✔	6 3.1 4.2 5.1 2.3 7.4 5.9	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	Input List [3.1, 4.2, 5.1, 2.3, 7.4, 5.9] pivot: 3.1 pivot: 5.1 pivot: 7.4 Sorted List [2.3, 3.1, 4.2, 5.1, 5.9, 7.4]	✔

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 for Bad Character Heuristic of Boyer Moore String Matching Algorithm

For example:

Input	Result
ABAAAABCD ABC	Pattern occur at shift = 5

Answer: (penalty regime: 0 %)

Reset answer

```
NO_OF_CHARS = 256
def badCharHeuristic(string, size):
    ########### Add your Code Here ###############
    badChar = [-1] * NO_OF_CHARS
    for i in range(size):
        badChar[ord(string[i])] = i
    return badChar
def search(txt, pat):
    m = len(pat)
    n = len(txt)
    badChar = badCharHeuristic(pat, m)
    s = 0
    while(s <= n-m):
        j = m-1
        while j>=0 and pat[j] == txt[s+j]:
            j -= 1
        if j<0:
            print("Pattern occur at shift = {}".format(s))
            s += (m-badChar[ord(txt[s+m])] if s+m<n else 1)
        else:
            s += max(1, j-badChar[ord(txt[s+j])])
def main():
```

	Input	Expected	Got	
✔	ABAAAABCD ABC	Pattern occur at shift = 5	Pattern occur at shift = 5	✔

Passed all tests! ✔

Correct

Marks for this submission: 20.00/20.00.