Started on	Saturday, 26 April 2025, 10:05 AM
State	Finished
Completed on	Saturday, 26 April 2025, 10:46 AM
Time taken	40 mins 57 secs
Grade	**80.00** out of 100.00

Question **1**

Correct

Mark 20.00 out of 20.00

Write a python program to implement merge sort using iterative approach on the given list of values.

For example:

Test	Input	Result

`Merge_Sort(S)`	6
	4
	2
	3
	1
	6
	5
		`The Original array is:  [4, 2, 3, 1, 6, 5]`
		`Array after sorting is:  [1, 2, 3, 4, 5, 6]`

`Merge_Sort(S)`	5
	2
	6
	4
	3
	1
		`The Original array is:  [2, 6, 4, 3, 1]`
		`Array after sorting is:  [1, 2, 3, 4, 6]`

Answer: (penalty regime: 0 %)

```python
def Merge_Sort(S):
    s=len(S)
    if(s>1):
        m=s//2
        l=S[:m]
        r=S[m:]
        Merge_Sort(l)
        Merge_Sort(r)
        i=0
        j=0
        k=0
        l_s=len(l)
        r_s=len(r)
        while(i<l_s and j<r_s):
            if(l[i]<r[j]):
                S[k]=l[i]
                i+=1
            else:
                S[k]=r[j]
                j+=1
            k+=1
        while(i<l_s):
```

	Test	Input	Expected	Got	

✓	`Merge_Sort(S)`	6
		4
		2
		3
		1
		6
		5
			`The Original array is:  [4, 2, 3, 1, 6, 5]`
			`Array after sorting is:  [1, 2, 3, 4, 5, 6]`
				`The Original array is:  [4, 2, 3, 1, 6, 5]`
				`Array after sorting is:  [1, 2, 3, 4, 5, 6]`
					✓

	Test	Input	Expected	Got	

✓	`Merge_Sort(S)`	5
		2
		6
		4
		3
		1
			`The Original array is:  [2, 6, 4, 3, 1]`
			`Array after sorting is:  [1, 2, 3, 4, 6]`
				`The Original array is:  [2, 6, 4, 3, 1]`
				`Array after sorting is:  [1, 2, 3, 4, 6]`
					✓

✓	`Merge_Sort(S)`	4
		3
		5
		6
		1
			`The Original array is:  [3, 5, 6, 1]`
			`Array after sorting is:  [1, 3, 5, 6]`
				`The Original array is:  [3, 5, 6, 1]`
				`Array after sorting is:  [1, 3, 5, 6]`
					✓

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 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):
    i=0
    j=0
    while(i<len(s1) and j<len(s2)):
        if(s1[i]==s2[j]):
            i+=1
            j+=1
        else:
            i=i-j+1
            j=0
        if(j>=len(s2)):
            return i-len(s2)
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 **3**

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):
        if(vertex==self.start and len(self.cycle)==N+1):
            self.hasCycle=True
            self.displayCycle()
            return
        for i in range(len(vertices)):
            if(adjacencyM[vertex][i]==1 and visited[i]==0):
                nbr=i
                visited[nbr]=1
                self.cycle.append(nbr)
                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 **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

```
def KMPSearch(pat, txt):

############## Add your code here #################
    M=len(pat)
    N=len(txt)
    lps=[0]*M
    j=0
    computeLPSArray(pat,M,lps)
    i=0
    while(N-i)>=(M-j):
        if(pat[j]==txt[i]):
            i+=1
            j+=1
        if(j==M):
            print("Found pattern at index "+str(i-j))
            j=lps[j-1]
        if(i<N and pat[j]!=txt[i]):
            if(j!=0):
                j=lps[j-1]
            else:
                i+=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**

Incorrect

Mark 0.00 out of 20.00

Write a python program to find minimum steps to reach to specific cell in minimum moves by knight.

Answer: (penalty regime: 0 %)

Reset answer

```
1  class cell:
2
3      def __init__(self, x = 0, y = 0, dist = 0):
4          self.x = x
5          self.y = y
6          self.dist = dist
7
8  def isInside(x, y, N):
9      if (x >= 1 and x <= N and
10         y >= 1 and y <= N):
11         return True
12     return False
13 def minStepToReachTarget(knightpos,
14                         targetpos, N):
15
16     # add your code here
17
18 if __name__=='__main__':
19     N = 30
20     knightpos = [1, 1]
21     targetpos = [30, 30]
22     print(minStepToReachTarget(knightpos,
```

Syntax Error(s)

```
Sorry: IndentationError: expected an indented block (__tester__.python3, line 18)
```

Incorrect

Marks for this submission: 0.00/20.00.