Started on	Monday, 28 April 2025, 3:13 PM
State	Finished
Completed on	Monday, 28 April 2025, 3:56 PM
Time taken	43 mins 2 secs
Grade	**100.00** 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):
############## Add your code here #############
    m=len(s1)
    n=len(s2)
    for i in range(m-n+1):
        j=0
        while j<n and s1[i+j]==s2[j]:
            j+=1
        if j==n:
            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! ✔

Marks for this submission: 20.00/20.00.

Question **2**

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])])
```

	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.

Question **3**

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

```
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):
    ######################### Add your code here ############################3
    if step_count==BOARD_SIZE**2+1:
        return True
    for step in STEPS:
        x_new=x+step[0]
        y_new=y+step[1]
        if is_safe(x_new,y_new):
            board[x_new][y_new]=step_count
            if solve_knights_tour(x_new, y_new, step_count+1):
                return True
            board[x_new][y_new]=0
    return False

def is_safe(x, y):
    return 0 <= x < BOARD_SIZE and 0 <= y < BOARD_SIZE and board[x][y] == 0

```

	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 **4**

Correct

Mark 20.00 out of 20.00

Write a python program to convert the given decimal number to binary number using recursive function.

For example:

Input	Result
10	1010
15	1111

Answer: (penalty regime: 0 %)

```
def decimal_to_binary(n):
    if n == 0:
        return ''
    else:
        return decimal_to_binary(n // 2) + str(n % 2)


decimal_number = int(input())

# Special case for 0
if decimal_number == 0:
    print(0)
else:
    binary_number = decimal_to_binary(decimal_number)
    print(binary_number)

```

	Input	Expected	Got	
✔	10	1010	1010	✔
✔	15	1111	1111	✔
✔	8	1000	1000	✔
✔	6	110	110	✔

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):
    ######################### Add your Code here ##########################
    path=[-1]*N
    path[0]=0
    def is_hamiltonian(path,pos):
        if pos==N:
            return True
        for v in range(N):
            if adj[path[pos-1]][v]==1 and v not in path:
                path[pos]=v
                if is_hamiltonian(path,pos+1):
                    return True
                path[pos]=-1
        return False
    if not is_hamiltonian(path,1):
        return False
    return True
adj = [ [ 0, 1, 1, 1, 0 ] ,
        [ 1, 0, 1, 0, 1 ],
        [ 1, 1, 0, 1, 1 ],
        [ 1, 0, 1, 0, 0 ] ]

```

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

Passed all tests! ✔

Marks for this submission: 20.00/20.00.