

<b>Started on</b>	Friday, 25 October 2024, 3:51 PM
<b>State</b>	Finished
<b>Completed on</b>	Friday, 25 October 2024, 3:59 PM
<b>Time taken</b>	7 mins 42 secs
<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 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

```
1 NO_OF_CHARS = 256
2 def badCharHeuristic(string, size):
3     ##### Add your Code Here #####
4
5     badChar = [-1] * NO_OF_CHARS
6     for i in range(size):
7         badChar[ord(string[i])] = i
8     return badChar
9
10
11 def search(txt, pat):
12     m = len(pat)
13     n = len(txt)
14     badChar = badCharHeuristic(pat, m)
15     s = 0
16     while(s <= n-m):
17         j = m-1
18         while j>=0 and pat[j] == txt[s+j]:
19             j -= 1
20         if j<0:
21             print("Pattern occur at shift = {}".format(s))
22             s += (m-badChar[ord(txt[s+m])] if s+m<n else 1)
```

	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 **2**

Correct

Mark 20.00 out of 20.00

Write a python program to check whether Hamiltonian path exists in the given graph.

**For example:**

Test	Result
Hamiltonian_path(adj, N)	YES

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

Reset answer

```

1
2 def Hamiltonian_path(adj, N):
3     ##### Add your Code here #####
4     dp = [[False for i in range(1 << N)] for j in range(N)]
5     for i in range(N):
6         dp[i][1 << i]=True
7     for i in range(1 << N):
8         for j in range(N):
9             if ((i & (1 << j))!=0):
10                for k in range(N):
11                    if((i & (1 << k)) != 0 and
12                        adj[k][j] == 1 and
13                        j != k and
14                        dp[k][i ^ (1 << j)]):
15                        dp[j][i]=True
16                        break
17     for i in range(N):
18         if (dp[i][(1 << N)-1]):
19             return True
20     return False
21
22

```

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

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

Question **3**

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.

## Question 4

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 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     ##### Add your Code here #####
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
16 def preprocess_case2(shift, bpos, pat, m):
17     j = bpos[0]
18     for i in range(m + 1):
19         if shift[i] == 0:
20             shift[i] = j
21         if i == j:
22             j = bpos[j]

```

	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 Saveetha veetha	pattern occurs at shift = 2 pattern occurs at shift = 22	pattern occurs at shift = 2 pattern occurs at shift = 22	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.

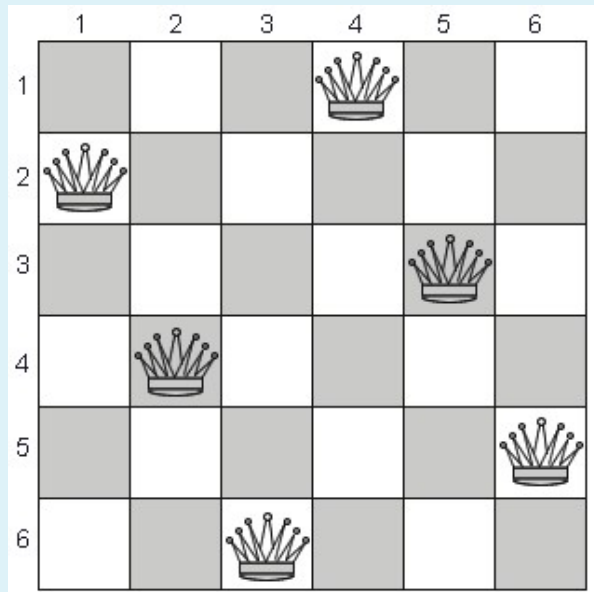
# Question 5

Correct

Mark 20.00 out of 20.00

You are given an integer **N**. For a given **N x N** chessboard, find a way to place '**N**' queens such that no queen can attack any other queen on the chessboard.

A queen can be attacked when it lies in the same row, column, or the same diagonal as any of the other queens. You need to find **one such configuration**.



**Note :**

Get the input from the user for N . The value of N must be from 1 to 6

If solution exists Print a binary matrix as output that has 1s for the cells where queens are placed

If there is no solution to the problem print "Solution does not exist"

For example:

Input	Result
6	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0

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

```

1 global N
2 N = int(input())
3
4 def printSolution(board):
5     for i in range(N):

```



	Input	Expected	Got	
✓	2	Solution does not exist	Solution does not exist	✓
✓	3	Solution does not exist	Solution does not exist	✓
✓	6	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	✓

Passed all tests! ✓

Correct

Marks for this submission: 20.00/20.00.