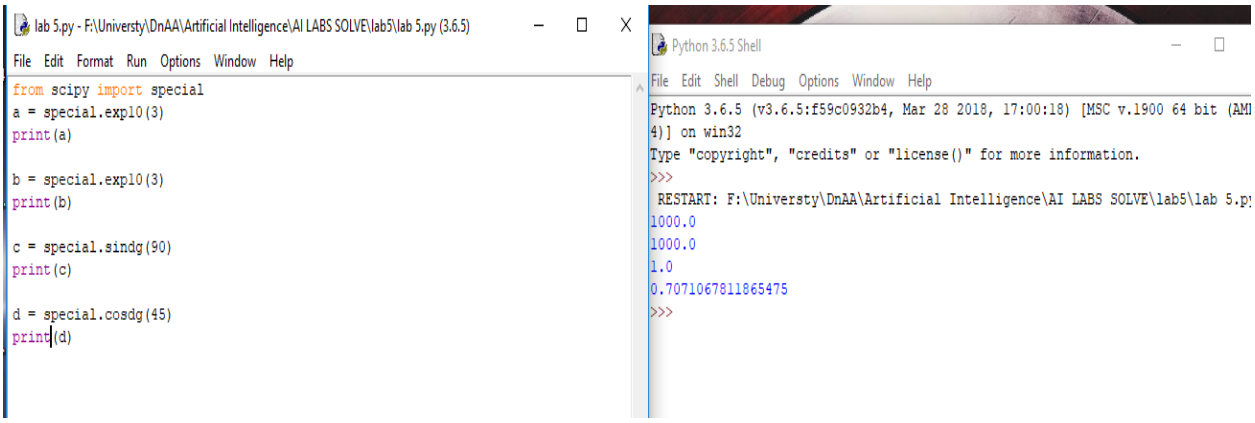
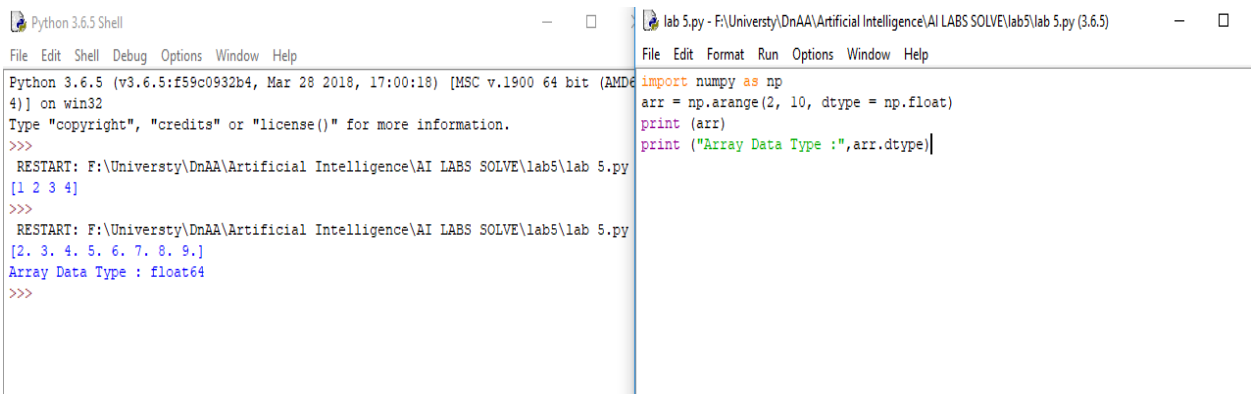
Name : Zaryab

Roll Num : BSCS/S17/0109

Lab 5

Q1- import package SciPy and implement 2 code examples illustrating usage of three different functions of this package.





Q2- Implement and understand 8 – Queen Algorithm and code.

Python 3.8.2 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (In
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: F:/Universty/DnAA/Artificial Intelligence/AI LABS SOLVE/lab5/lab 5.py
How big is your chess board?8
[(1, 1), (2, 5), (3, 8), (4, 6), (5, 3), (6, 7), (7, 2), (8, 4)]
[(1, 1), (2, 6), (3, 8), (4, 3), (5, 7), (6, 4), (7, 2), (8, 5)]
[(1, 1), (2, 7), (3, 4), (4, 6), (5, 8), (6, 2), (7, 5), (8, 3)]
[(1, 1), (2, 7), (3, 5), (4, 8), (5, 2), (6, 4), (7, 6), (8, 3)]
[(1, 2), (2, 4), (3, 6), (4, 8), (5, 3), (6, 1), (7, 7), (8, 5)]
[(1, 2), (2, 5), (3, 7), (4, 1), (5, 3), (6, 8), (7, 6), (8, 4)]
[(1, 2), (2, 5), (3, 7), (4, 4), (5, 1), (6, 8), (7, 6), (8, 3)]
[(1, 2), (2, 6), (3, 1), (4, 7), (5, 4), (6, 8), (7, 3), (8, 5)]
[(1, 2), (2, 6), (3, 8), (4, 3), (5, 1), (6, 4), (7, 7), (8, 5)]
[(1, 2), (2, 7), (3, 3), (4, 6), (5, 8), (6, 5), (7, 1), (8, 4)]
[(1, 2), (2, 7), (3, 5), (4, 8), (5, 1), (6, 4), (7, 6), (8, 3)]
[(1, 2), (2, 8), (3, 6), (4, 1), (5, 3), (6, 5), (7, 7), (8, 4)]
[(1, 3), (2, 1), (3, 7), (4, 5), (5, 8), (6, 2), (7, 4), (8, 6)]
[(1, 3), (2, 5), (3, 2), (4, 8), (5, 1), (6, 7), (7, 4), (8, 6)]
[(1, 3), (2, 5), (3, 2), (4, 8), (5, 6), (6, 4), (7, 7), (8, 1)]
[(1, 3), (2, 5), (3, 7), (4, 1), (5, 4), (6, 2), (7, 8), (8, 6)]
[(1, 3), (2, 5), (3, 8), (4, 4), (5, 1), (6, 7), (7, 2), (8, 6)]
[(1, 3), (2, 6), (3, 2), (4, 5), (5, 8), (6, 1), (7, 7), (8, 4)]
[(1, 3), (2, 6), (3, 2), (4, 7), (5, 1), (6, 4), (7, 8), (8, 5)]
[(1, 3), (2, 6), (3, 2), (4, 7), (5, 5), (6, 1), (7, 8), (8, 4)]
[(1, 3), (2, 6), (3, 4), (4, 1), (5, 8), (6, 5), (7, 7), (8, 2)]
[(1, 3), (2, 6), (3, 4), (4, 2), (5, 8), (6, 5), (7, 7), (8, 1)]
[(1, 3), (2, 6), (3, 8), (4, 1), (5, 4), (6, 7), (7, 5), (8, 2)]
[(1, 3), (2, 6), (3, 8), (4, 1), (5, 5), (6, 7), (7, 2), (8, 4)]
[(1, 3), (2, 6), (3, 8), (4, 2), (5, 4), (6, 1), (7, 7), (8, 5)]
[(1, 3), (2, 7), (3, 2), (4, 8), (5, 5), (6, 1), (7, 4), (8, 6)]
[(1, 3), (2, 7), (3, 2), (4, 8), (5, 6), (6, 4), (7, 1), (8, 5)]
[(1, 3), (2, 8), (3, 4), (4, 7), (5, 1), (6, 6), (7, 2), (8, 5)]
[(1, 4), (2, 1), (3, 5), (4, 8), (5, 2), (6, 7), (7, 3), (8, 6)]
[(1, 4), (2, 1), (3, 5), (4, 8), (5, 6), (6, 3), (7, 7), (8, 2)]
[(1, 4), (2, 2), (3, 5), (4, 8), (5, 6), (6, 1), (7, 3), (8, 7)]
[(1, 4), (2, 2), (3, 7), (4, 3), (5, 6), (6, 8), (7, 1), (8, 5)]
[(1, 4), (2, 2), (3, 7), (4, 3), (5, 6), (6, 8), (7, 5), (8, 1)]
[(1, 4), (2, 2), (3, 7), (4, 5), (5, 1), (6, 8), (7, 6), (8, 3)]
```

lab 5.py - F:/Universty/DnAA/Artificial Intelligence/AI LABS SOLVE/lab5/lab 5.py (3.8.2)

File Edit Format Run Options Window Help

```
from itertools import permutations, combinations

text = input('How big is your chess board?')
n = int(text)
x = range(1, n+1)

def is_diagonal(point1, point2):
    x1 = point1[0]
    y1 = point1[1]
    x2 = point2[0]
    y2 = point2[1]
    gradient = (y2-y1)/(x2-x1)
    if gradient == -1 or gradient == 1:
        return(True)
    else:
        return(False)

list_of_permutations = []

for permuation in permutations(range(1, n+1)):
    y = permuation
    all_permutations = list(zip(x,y))
    list_of_permutations.append(all_permutations)

for possible_solution in list_of_permutations:
    solutions = []
    for piece1, piece2 in combinations(possible_solution, 2):
        solutions.append(is_diagonal(piece1, piece2))

    if True not in solutions:
        print(possible_solution)
```