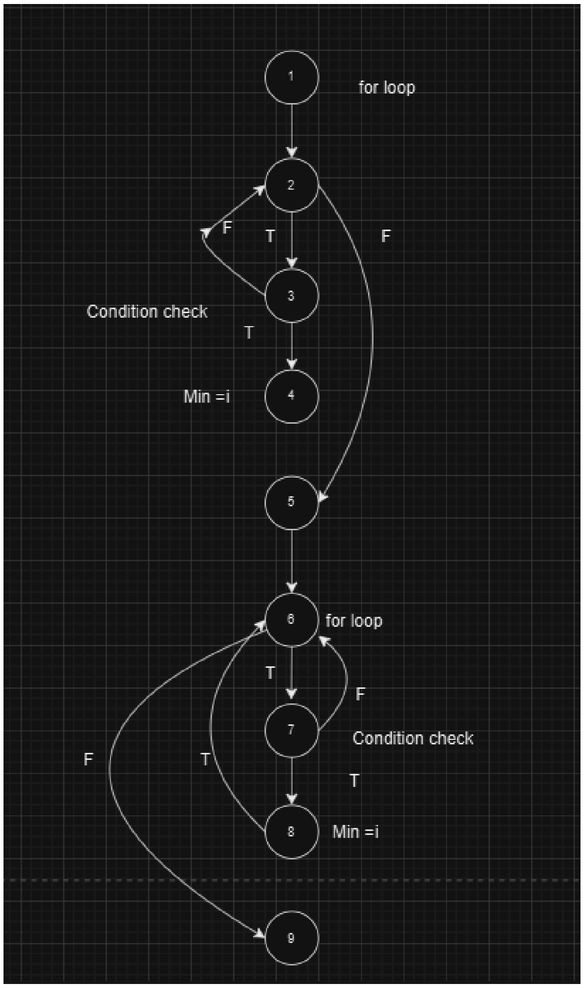
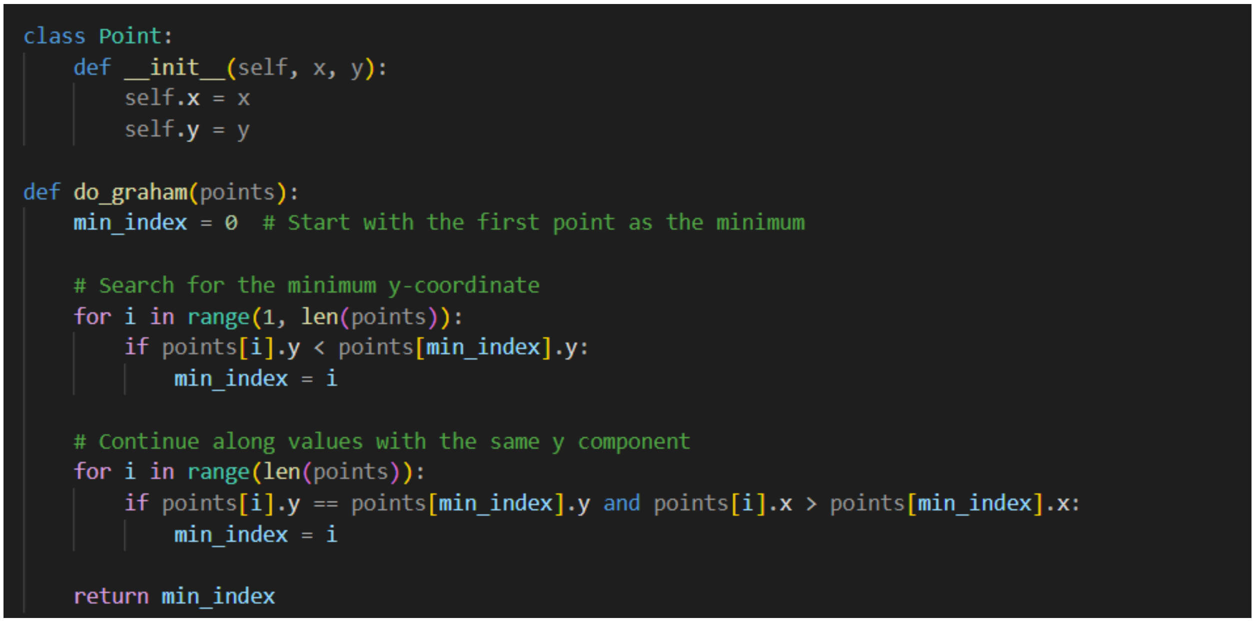
SE-Lab9
202201524

Control flow graph :

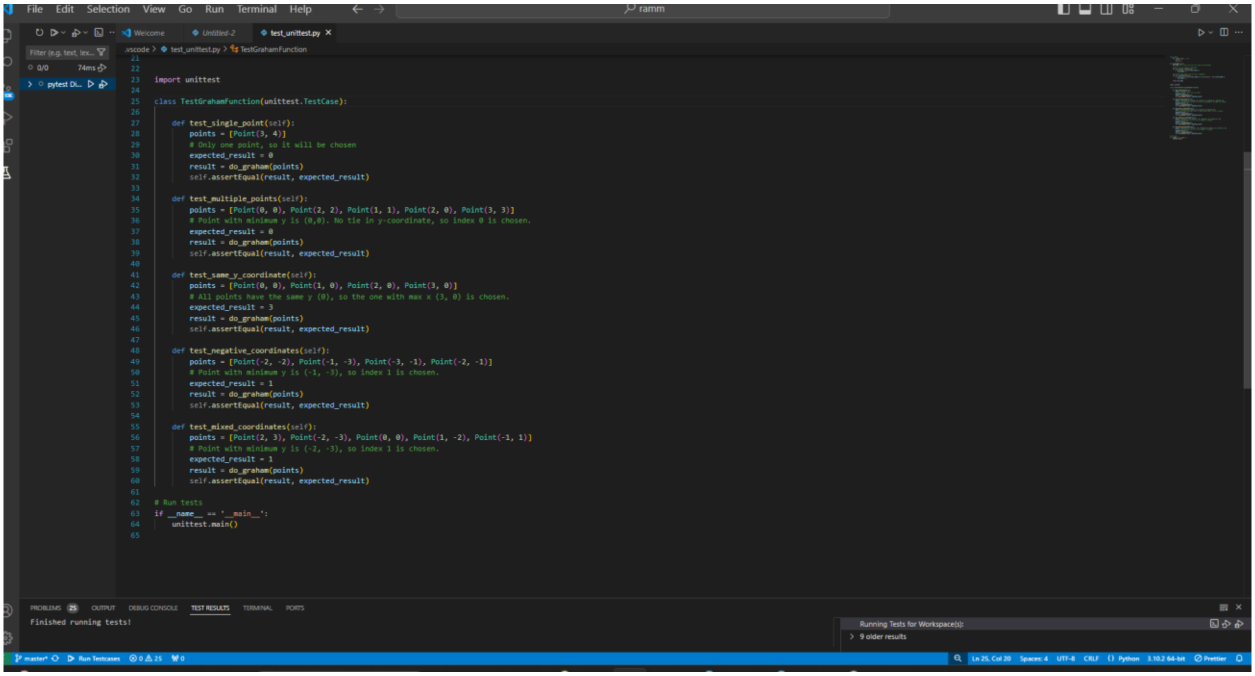


Code in python:



```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

def do_graham(points):
    min_index = 0  # Start with the first point as the minimum

    # Search for the minimum y-coordinate
    for i in range(1, len(points)):
        if points[i].y < points[min_index].y:
            min_index = i

    # Continue along values with the same y component
    for i in range(len(points)):
        if points[i].y == points[min_index].y and points[i].x > points[min_index].x:
            min_index = i

    return min_index
```

TestCase :



```python


import unittest

class TestGrahamFunction(unittest.TestCase):

    def test_single_point(self):
        points = [Point(3, 4)]
        # Only one point, so it will be chosen
        expected_result = 0
        result = do_graham(points)
        self.assertEqual(result, expected_result)

    def test_multiple_points(self):
        points = [Point(0, 0), Point(2, 2), Point(1, 1), Point(2, 0), Point(3, 3)]
        # Point with minimum y is (0,0). No tie in y-coordinate, so index 0 is chosen.
        expected_result = 0
        result = do_graham(points)
        self.assertEqual(result, expected_result)

    def test_same_y_coordinate(self):
        points = [Point(0, 0), Point(1, 0), Point(2, 0), Point(3, 0)]
        # All points have the same y (0), so the one with max x (3, 0) is chosen.
        expected_result = 3
        result = do_graham(points)
        self.assertEqual(result, expected_result)

    def test_negative_coordinates(self):
        points = [Point(-2, -2), Point(-1, -3), Point(-3, -1), Point(-2, -1)]
        # Point with minimum y is (-1, -3), so index 1 is chosen.
        expected_result = 1
        result = do_graham(points)
        self.assertEqual(result, expected_result)

    def test_mixed_coordinates(self):
        points = [Point(2, 3), Point(-2, -3), Point(0, 0), Point(1, -2), Point(-1, 1)]
        # Point with minimum y is (-2, -3), so index 1 is chosen.
        expected_result = 1
        result = do_graham(points)
        self.assertEqual(result, expected_result)

# Run tests
if __name__ == '__main__':
    unittest.main()

```

```
Finished running tests!
```