Grading Guidelines for Assignments 11 and 12 (January 21)

For partial marking in code parts, use your judgement, but in any part, if they have something looking more or less ok, they should get some marks. So if some part has only 1 mark, deduct 0.5 and not 1 unless they are totally off.

Assignment 11 (Total Marks = 10):

- Output: 2 marks (0.5 marks each binary)
  - Test case 1:
    - Rectangle: Top-left (3, 12), bottom right (10,10)
    - Line segment: (5,11) and (12,11)
    - Should print 1
  - Test case 2:
    - Rectangle: Top-left (3, 12), bottom right (10,10)
    - Line segment: (1,13) and (12,13)
    - Should print 0
  - Test case 3:
    - Rectangle: Top-left (3, 12), bottom right (10,10)
    - Line segment: (1, 11) and (12,11)
    - Should print 2
  - Test case 4:
    - Rectangle: Top-left (3, 12), bottom right (10,10)
    - Line segment: (1, 12) and (12,12)
    - Should either print 0 or in some form mention infinite (can be a message)
- Code: 8 marks
  - Defining the structures: 1 mark
  - Main function: 2 marks
    - Declaring structure variables and reading in values in them: 1 mark
    - Calling Intersects() correctly and printing the number of intersections: 1 mark
  - Intersects function: 5 marks
    - Works for horizontal line segment fully above (and out of) rectangle and fully below and (out of) : 1 mark
    - Same as above for vertical line segment fully to the left and fully to the right (and out) of rectangle: 1 mark
    - Works for horizontal line segments with one end point inside and one endpoint outside (check for outside point both towards left and right)): 1 mark
    - Same as above but for vertical line segment: 1 mark
    - Works for line segments with both points outside but cutting the rectangle (2 intersection points): 1 mark (0.5 each for horizontal and vertical)

        - They can have conditions to test the above in any way. But make sure all cases are tested. I suggest just form an example of your own for each case and see what happens for each. But feel free to do it your way.

Assignment 12 (Total Marks = 10):

- Output (binary marking: either 0.5 or 0 for each test case): 2 marks
    - Test Case 1: A = “abc123”, B = “abc”
        - Should say A is not a prefix of B (no marks if they say B is prefix of A, that was not the question)
    - Test Case 1: A = “abc”, B = “abc123”
        - Should say A is a prefix of B
    - Test Case 1: A = “abcd12”, B = “abc123”
        - Should say A is not a prefix of B
    - Test case 4: A = “abcd”, B = “abcd”
        - Should say A is a prefix of B
- Code: 8 marks
    - Main function: 2 marks
        - Reading in A, B with %s and finding length– 1 mark
        - Calling IsPrefix and printing message based on return value – 1 mark
        - Deduct 1 if they do loops and such to find prefix of substrings etc. etc., or read in anything else other than the two strings
        - Deduct 0.5 if they check the 1st character in main() and then call IsPrefix(). The slide was very clear on what main() should do.
    - Is Prefix function: 6 marks
        - Handles the case when h1-l1 is > than h2 – l2, (should return 0): 2 mark
            - They can do it explicitly first or in the base condition
        - Correct base condition (h1=l1 or equivalent) and return value after check: 2 mark
        - Correct recursive call and storing the return value: 2 mark
        - Deduct 2 if they print anything from inside the function. Clearly told not to do so.
        - If the recursive formulation is not the simple one (they can check either first or last character), deduct 2 (Well, I cannot think of anything, but in case). Definitely deduct 2 if they have loops in there.