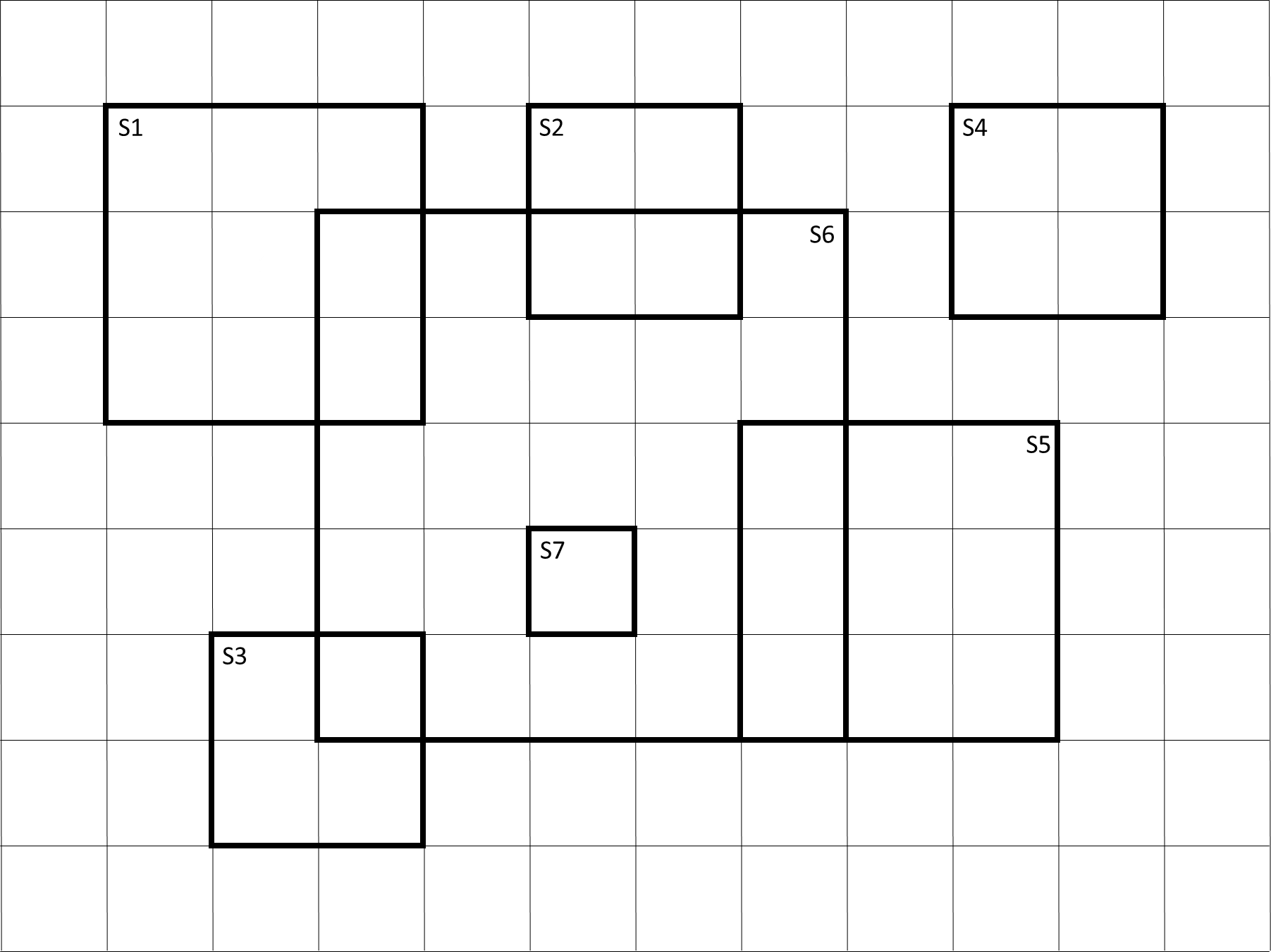
**Problem 0: Visual Studio (0 points)**

Download and install Microsoft Visual Studio. You are entitled to a free copy using DePaul’s MSDNAA license. “The MSDN Academic Alliance (MSDNAA) is a special program to provide academic institutions, faculty, and students with leading-edge tools for students.” You can find more information about the MSDNAA on the CDM website (<www.cdm.depaul.edu/Current%20Students/Pages/MSDNAA.aspx>).

If you are unfamiliar with Visual studio, Microsoft has some useful YouTube videos regarding the installation and option:  <https://www.youtube.com/watch?v=cjVx8xbxt80>

**Problem 1: Squares (100 points)**

A square can be defined by 3 ints: x, the horizontal position of the lower left corner; y, the vertical position of the lower left corner; and s, the height and width of the square. These three ints can be stored in a list. For example, [8,1,3] would represent the square S1 in the figure below.



Write a function that accepts two lists, each representations of a square. Compute the area of the overlap of the two squares. If the squares do not overlap, return 0. Test your overlap function with the code below. Do not alter this code. Include this code at the end of your submission.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, “I have not given or received any unauthorized assistance on this assignment.” Each function must include a docstring and be commented appropriately.

totalScore = 0

S1 = [1,5,3]

S2 = [5,6,2]

S3 = [2,1,2]

S4 = [9,6,2]

S5 = [7,2,3]

S6 = [3,2,5]

S7 = [5,3,1]

#---------- ---------- ---------- ---------- ----------

print( "Test 1: " + str(S1) + str(S6) )

print( "Correct Answer: 2" )

r1 = overlap(S1,S6)

r2 = overlap(S6,S1)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 2:

s1 = s1 + 1

if r2 == 2:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 2: " + str(S2) + str(S6) )

print( "Correct Answer: 2" )

r1 = overlap(S2,S6)

r2 = overlap(S6,S2)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 2:

s1 = s1 + 1

if r2 == 2:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 3: " + str(S3) + str(S6) )

print( "Correct Answer: 1" )

r1 = overlap(S3,S6)

r2 = overlap(S6,S3)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 1:

s1 = s1 + 1

if r2 == 1:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 4: " + str(S4) + str(S6) )

print( "Correct Answer: 0" )

r1 = overlap(S4,S6)

r2 = overlap(S6,S4)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 0:

s1 = s1 + 1

if r2 == 0:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 5: " + str(S5) + str(S6) )

print( "Correct Answer: 3" )

r1 = overlap(S5,S6)

r2 = overlap(S6,S5)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 3:

s1 = s1 + 1

if r2 == 3:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 6: " + str(S6) + str(S6) )

print( "Correct Answer: 25" )

r1 = overlap(S6,S6)

r2 = overlap(S6,S6)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 25:

s1 = s1 + 1

if r2 == 25:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print( "Test 7: " + str(S7) + str(S6) )

print( "Correct Answer: 1" )

r1 = overlap(S7,S6)

r2 = overlap(S6,S7)

print( "Result 1: " + str(r1) )

print( "Result 2: " + str(r2) )

s1 = 0

if r1 == 1:

s1 = s1 + 1

if r2 == 1:

s1 = s1 + 1

print( "Score: " + str(s1) )

print()

totalScore = totalScore + s1

#---------- ---------- ---------- ---------- ----------

print ( "Total Score: " + str(totalScore) )

print ( "Percentage: " + str(100\*totalScore/14) )

**Problem 2: Happy Primes**

1. A prime number (or a prime) is a natural number greater than 1 that cannot be formed by multiplying two smaller natural numbers. Write a function that returns true if an int is prime. If the int is not prime or if it is an invalid input return false. Write your code efficiently.
2. “Happy” has many definitions (even in mathematics). For our purposes, a happy number is a number defined by the following process: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number either equals 1 (where it will stay), or it loops endlessly in a cycle that does not include 1. Those numbers for which this process ends in 1 are happy numbers, while those that do not end in 1 are unhappy numbers (or sad numbers). For example, 19 is happy, as the associated sequence is:  
   * 12 + 92 = 82
   * 82 + 22 = 68
   * 62 + 82 = 100
   * 12 + 02 + 02 = 1

Write a function that returns true if an int is happy.

1. Write a function that returns true if an int is a happy prime, the int is both happy and a prime.
2. (25 points) Print to the console the first 20 primes.
3. (25 points) Print to the console the first 20 happy numbers.
4. (25 points) Print to the console the first 20 happy primes.
5. (25 points) Print to the console the first 20 sad primes.

Note: Please consider the efficiency of your code.

Submission: Submit a single .py file containing all the code to the D2L. Do not zip or archive the file. Your code must include comments at the top including your name, date and the honor statement, “I have not given or received any unauthorized assistance on this assignment.” Each function must include a docstring and be commented appropriately.