Problem Solving Session

- The remainder of today's class will comprise the ***problem solving session*** (***PSS***).
- Your instructor will divide you into ***teams of 3 or 4 students***.
- Each team will ***work together*** to solve the following problems over the course of ***20-30 minutes***.
  - You may work on paper, a white board, or digitally as determined by your instructor.
  - You will submit your solution by pushing it to GitHub before the end of class.
- Your instructor will go over the solution before the end of class.
- If there is any time remaining, you will begin work on your homework assignment.

Class participation is a significant part of your grade (20%). This includes in class activities and the problem solving session.

Your Course Assistants will grade your participation by verifying that you pushed your solutions before the end of the class period each day.

Problem Solving Team Members

Record the name of each of your problem solving team members here.

Do not forget to ***add every team member's name***! Your instructor (or course assistant) may or may not use this to determine whether or not you participated in the problem solving session.

Ryan Robison
Jackson Palmer

Problem Solving 1

The `turtle.write` command writes a string immediately above the turtle cursor. Assuming the string is center justified, write the python code to ensure a letter is written in the center of the bounding box. The turtle will start at the lower left corner of the box.

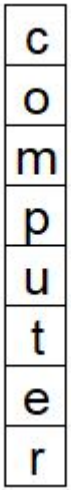
Example letter

```
Move turtle forward half box length
Draw the letter directly above
```

```
Move turtle to start x,y
Make a boxsize variable
Make an iteration variable
For st in range(“computer”)
     turtle.write(st)
     turtle.goto(x,y-boxsize*iteration)
     Increase iteration
```

Problem Solving 2

Each word in the our crossword solution maker can be written either **vertically** or **horizontally**. Assume there is already a move function that will `move` the turtle to the start of any letter on the drawing grid. Write the a function that will write all the letters in the word `"computer"` vertically, starting at a specified x,y coordinate.



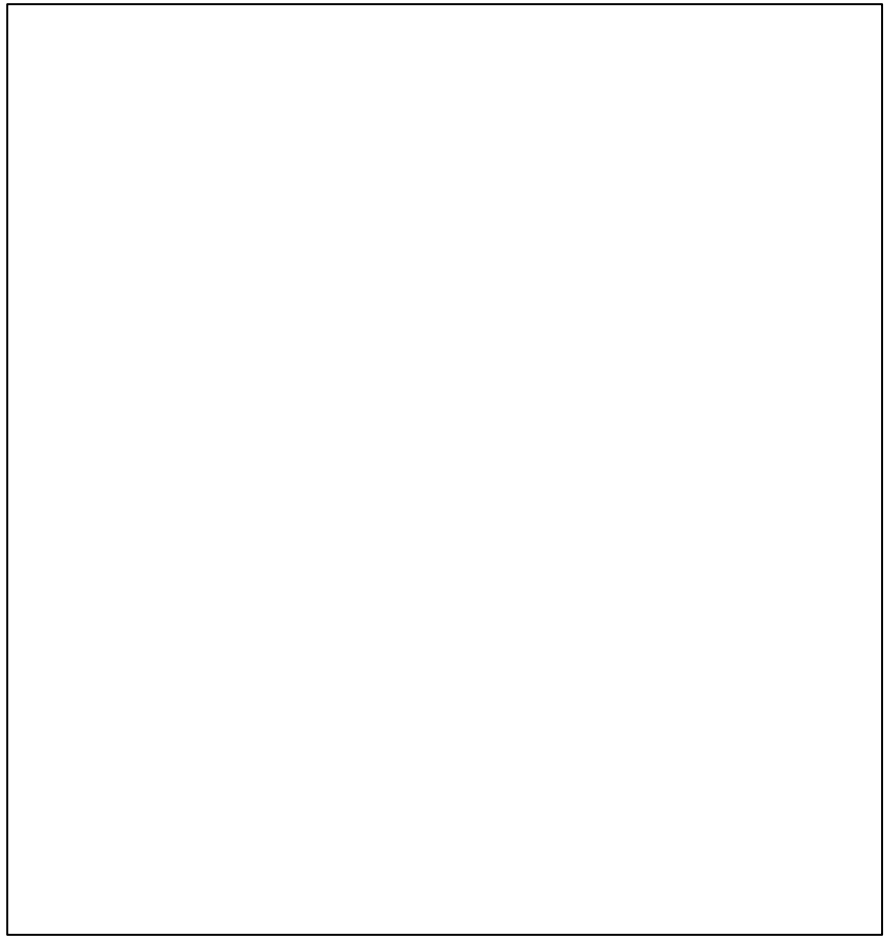
Your code does not need to draw the outer boxes

Problem Solving 3

Words must fit within a 20 x 20 grid. Write the code necessary to validate that a horizontal word would fit within the grid. The location of the first letter in the word is given in X,Y coordinates where (0,0) is the upper left element and (19,19) is the lower right. The word itself is also provided.

Example: `validate(8, 12, "GCIS-123")`

```
#first you need to take the length of the
string to see if it will fit inside the
grid. Then declare it as a new int

stringLength = len(string)

#create an if statement comparing the
string to the grid

If (stringLength > x):
     Return('string will not fit')

Elif (stringLength <= x):
          Return ('string will fit')

The same steps would have to be used to
compare the vertical height of the string.
```



Problem Solving 4

Words will be provided by the user as strings in the following format.

```
<X> <Y> <Word> <'H'|'V'>
```

For example:

```
2 12 tremendous V
```

where 'H' or 'V' indicates the word is to be written horizontally or vertically. Write the code necessary to read a string from the user and then place the correct values into the 4 variables listed below:

```
x_cor         # int
y_cor         # int
word          # string
is_horizontal # boolean
```