

Fall 2020 COP 2930 Final Exam Part B Solution

1) (10 pts) Take the work from question 3 of part A and turn it into a function. Formally, complete the following function with the pre and post conditions given. Note: your function should have no print or input calls in it. If it does, you'll get minimal credit for this question.

Sample Solution

```
# Pre-condition: n is a positive integer
# Post-condition: returns the largest integer k such that
# k*k*k <= n.
def intcuberoot(n):

    x = 1

    while x*x*x <= n:
        x += 1

    return x-1
```

**Grading: 3 pts for having no input call in the function,
4 pts for copying the relevant functionality from their answer in part A
3 pts for both removing the print and adding the return**

2) (10 pts) You must call the following void function in your solution to this problem:

```
# Pre-condition: ch is a single character, n is a positive integer
# Post-condition: ch will be printed n times, followed by the
#                 newline character.
def printLine(ch, n)
```

An alternating triangle design is one where the rows alternate between printing '*' and '=', where each row has one more character than the previous row. The first row can start with either character and has exactly 1 character on it. Write a void function that takes in the number of rows and the starting character for the first row and prints the appropriate alternating triangle design. For example, the function call `printAltTri(5, '*')` would print the following:

```
*
==
***
====
*****
```

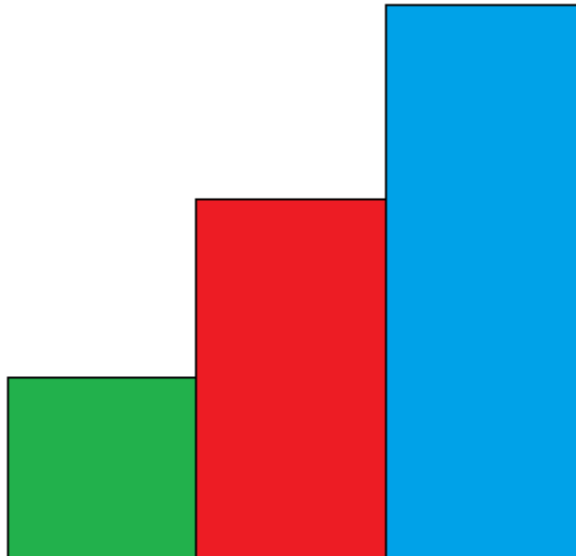
Your function **MUST MAKE CALLS** to the `printLine` function to earn full credit. Correct solutions that don't call the function will lose **SUBSTANTIAL CREDIT**.

Sample Solution

```
def printAltTri(numRows, ch):  
  
    for i in range(1, numRows+1):  
        printLine(ch, i)  
  
        if ch == '*':  
            ch = '='  
        else:  
            ch = '*'
```

Grading: 2 pts for no prints or inputs in the function
2 pts for having the correct function signature
2 pts for the outer loop
2 pts for the correct function call in the loop
2 pts for the character toggle (lots of ways to do this)

3) (10 pts) Use the Python turtle to draw the following design.



The coordinates of the bottom left corner of the green square are (0 , 0) and the length of each side of the green square is 100 pixels. The red rectangle has a width of 100 pixels and a height of 200 pixels. The blue rectangle has a width of 100 pixels and a height of 300 pixels. In your solution, please limit yourself to using the following turtle functions:

forward
penup
pendown
left
right

```
begin_fill
end_fill
fillcolor
```

Please use the default colors "green", "red" and "blue" that Python provides. **If you like, you can work on this one in IDLE before typing your response on the file you submit.**

Sample Solution

```
import turtle

turtle.fillcolor("green")
turtle.begin_fill()
for i in range(4):
    turtle.forward(100)
    turtle.left(90)
turtle.end_fill()

turtle.fillcolor("red")
turtle.forward(100)
turtle.begin_fill()

for i in range(2):
    turtle.forward(100)
    turtle.left(90)
    turtle.forward(200)
    turtle.left(90)
turtle.end_fill()

turtle.fillcolor("blue")
turtle.forward(100)
turtle.begin_fill()
for i in range(2):
    turtle.forward(100)
    turtle.left(90)
    turtle.forward(300)
    turtle.left(90)
turtle.end_fill()
```

Grading: 2 pts for each shape outline (so 6 pts total here), 1 pt for each fill management (so 3 pts total here), 1 pt total for the color assignments, don't take off 1 pt per incorrect line since there are more than 10 lines here. Don't take off any credit for lack of loops.