

CS 150: Intro to OOP, Spring 2012

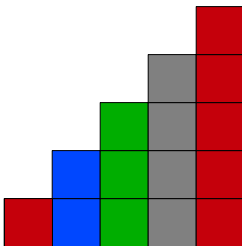
Assignment 4

Due via email by 11:59pm on February 20, 2012

For this assignment you must work individually on these problems. Please note the distinction made in our academic integrity policy between general course material and work which is submitted for this course. We consider the use of the Python language syntax and the `cs1graphics` package in the category of general course material which you may discuss freely. However, you must avoid any discussion of code which is specific to this assignment. You should not receive direct help from others, nor should you share your own source code with others.

You will be producing a set of 4 individual programs and one set of answers for the problems below.

1. Using the `cs1graphics` module, write a program that draws an n -level staircase of squares as shown below. Each column will be colored. The colors to use are red, blue, green, and gray in that order. For more than 4 columns repeat the colors. Prompt the user for n . For $n=5$ the staircase would look like this:



2. Write a program that allows the user to enter words, continuing until the user first enters a duplicate word. The duplicate word does not need to match in case. A sample of a running program is:

```
Start entering words -
kiwi
apple
banana
Apple
3 words were entered before seeing the duplicate "Apple"
```

3. An integer $k \geq 2$ is a *prime number* if it is not evenly divisible by any numbers in $\text{range}(2,k)$.
 - a) Build a list of all prime numbers less than or equal to a value n entered by a user. For each $2 \leq k \leq n$, test its primality by looping over $\text{range}(2,k)$ looking for a factor of k . Print the list of primes.
 - b) Optimize your program from part a) by only testing the primality of each new k by using previously found primes. If k is not divisible by a prime factor, it cannot be divisible by any other factor. Print the list of primes.

4. Write a function `pairSum(data, goal)` that returns `True` if the data contains two distinct elements whose sum equals the goal, and `False` otherwise.

For example, using `data = [5, 12, 3, 8, 12, 11, 2, 6]`,
then `pairSum(data, 14)` would return `True` because 12 and 2 sum to 14.
also `pairSum(data, 24)` would return `True` because 12 occurs twice in the list,
but `pairSum(data, 4)` would return `False` because there is only a single 2 in the list.

In the main part of the program, prompt the user for numbers to check and print `True` or `False` until the user enters a zero value or an empty line.

5. For this problem do not submit a program only the answers.

Carefully consider the following program:

```
answer = 1
if greeting.count('a') == 1:
    if 'o' in greeting:
        if greeting.endswith('o'):
            answer = 2
        else:
            answer = 3
    elif len(greeting) < 6:
        answer = 4
print answer, greeting
```

For each of the following greeting values, predict the output that would result.

```
adieu
aloha
bonjour
guten tag
hallo
molo
shalom
terve
```