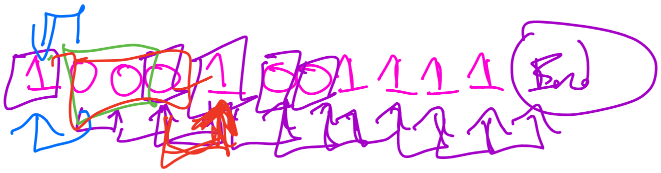
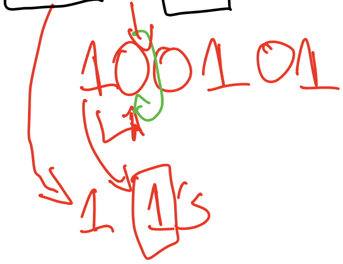
Lab Question Review



1	1's
3	0's
1	1's
2	0's
4	1
acc	#



In order to count the number of consecutive integers.

```
def count_consec(str_bin):
    acc = 0
    for i in range(1, len(str_bin)):
        if str_bin[i] == str_bin[i-1]:
            acc +=1
        else:
            print(acc, str_bin[i-1]+"'s")
            acc = 1
```

start at the first index so you can compare backwards.

print the accumulater, then print the str_bin representation @ index [i-1].

Add to the accumulater otherwise.

Class (Object Oriented Programming) Review

```
Class Shoe:
    def __init__(self, name, brand, sz=13):
        self.name = name
        self.brand = brand
        self.size = sz
```

we use self to manage our data within our class.

```
    def price(self):
        # we'll use the ord and add each character
        # to one another to obtain the price.
        acc = 0
        for char in self.name:
            acc+= ord(char)

        for char1 in self.brand:
            acc += ord(char1)

        return acc
```

we are using a function in our class, inside we have to call self in the function

```
    def __str__(self);
        return str(self.name) + " ," + str(self.brand) + " $" + str(Shoe.price(self))

# Str returns the string representation of your class.
# Classes can be incredibly useful when representing data.
# you can pretty much create classes for anything!
#In order to call the function, we use:
airforce1 = Shoe("AF1", "Nike", 12)
print(airforce1) # grabs the String representation of the function.
```

OUTPUT:

"AF1 , Nike ,$ (some number)"

```
Class Perfume:
    def __init__(self, name, shape, color, scent="lavender"):
        self.nm=name
        self.sh= shape
        self.col = color
        self.sc = scent
    def price(self):
        #multiply the length of the name by the length of the
        #scent
        return len(self.nm) * len(self.sc)
    def size(self):
        list =[('square', '2oz'), ('circle', '4oz')]
        for i in list:
            if self.shape == i[0]:
                return i[2]
    def __str__(self):
        return str(self.name)+", $"+str(Perfume.price(self)) + ",
            size"+ str( Perfume.size(self));

calyx= Perfume("Calyx", "Square", "Clear", "lavender")
print(calyx)
```