These problems were generated by TAs and instructors in previous semesters. They may or may not match the actual difficulty of problems on Exam1.

Algorithms and Abstraction

1. As you learned in lecture, computer science boils down to two main ideas: algorithms and abstraction.
   1) Identify/Define the difference between algorithms and abstraction, and provide an example for each.
   2) What are the 3 key components of a good algorithm?

Programming Basics

1. For each of the following Python expressions, identify the data type of the value it will evaluate to.
   1) 3
   2) 3.0
   3) "3"
   4) 3 < 3
   5) "3" + "3"
   6) 3.0 * 3.0
   7) 3 / 3

2. Suppose you are creating a few variables to describe yourself or another person. Name the data type that would **best** fit each variable description:
   1) Name
   2) Age
   3) Height (inches)
   4) Ability to vote
   5) Address
   6) Zip code
   7) Weight
   8) School you attend
   9) Ability to speak more than one language

Data Representation

1. Answer the following questions regarding number systems:
   1) What is a number system?
   2) How many values can a specific digit have in a decimal number? In a binary number?
   3) Let's say that we are representing length in imperial units. You are given inches, feet, and yards. How could you represent 10,000 inches in this system (show your work)? (hint: 12 inches=foot, 3 feet=yard 1760 feet=mile)

2. Complete the following conversions and show appropriate work:
   1) Convert 10111011 from binary to decimal
   2) Convert 11111111 from binary to decimal
   3) Convert 01101110 from binary to decimal
   4) Convert 57 from decimal to 8-bit binary
   5) Convert 63 from decimal to 8-bit binary
   6) Convert 88 from decimal to 8-bit binary

3. The following questions relate to interpreting binary numbers as abstracted types
   1) Answer the following questions relating to abstracting colors:
      a) How many bytes are needed to represent an RGB value?
      b) What are the highest and lowest values for any color?
      c) Convert the color purple to an RGB value. Provide the values in decimal and binary.
      d) Convert 11010101 11110000 00110011 to a color (You can use this link to determine what color is appears to be: https://www.w3schools.com/colors/colors_rgb.asp)
   2) Answer the following questions relating to abstracting text (ASCII table: http://www.asciitable.com/):
      a) Convert 00111010 00101001 to ASCII characters.
      b) Convert 'CMU cmu' into 8-bit binary (neglect the quotation marks).

Functions (Calls and Definitions)

1. Evaluate the print statements and state what the value of y and x are at the end of the function.

```
x = 15+1

def f():
    y = 12
    print(y * 2)
    print(y)
    y = "apple"
    print("bananas, 15-110, and",y)
    print(x+3)

f()
```

2. Consider the following function:

```
def helloThere(name):
    print("Hello there,", name, "!")
```

1) What will you see in the shell if you call print(helloThere("Stella")) ?
2) Does this function have side effect(s)? If so, state the side effect(s).
3) Does this function have a returned value? If so, state the returned value.
4) What do we call the statement helloThere("Stella")? What do we call the string "Stella" in that statement?

3. Consider the following algorithm for a function:

Algorithm:
1. Input a number n
2. If n is greater than or equal to 0 do the following, otherwise skip to step 3:
   a. If n is divisible by 10 then print out the value of n followed by " is cool number", otherwise skip to step 2b
   b. If step 2a did not occur, check if n is divisible by 4, and if so print out the value of n followed by " is a strange number", otherwise skip to step 2c
   c. Skip to step 4
3. Print the value of n followed by " is a negative number!!"
4. Print the string "ALL DONE"

Identify the argument(s), returned value, and side effect(s) of this algorithm. Then write the corresponding Python function called divisionCheck.

4. The function fruitCalculator has parameters percentApples, totalFruit, and farm. fruitCalculator calculates the number of apples harvested by the farm, prints the number of apples and the number of other fruit, and returns a string announcing the harvest of apples. For example, fruitCalculator(0.75, 400, "Carnegie Farms") would return "Carnegie Farms harvested 300 apples this year", and the following would be printed in the console.

300 apples were harvested!
100 other fruits were also harvested!

Write the function fruitCalculator according to the description above.

5. Suppose we had the following segment of code:

```
course = "15110"
grade = 95.0
def randomFunction(x, y):
    sum = x + y
    difference = x - y
    print(course)
    return (sum + difference) / 2
```

Give the scopes, either local or global, of the following variables: course, grade, sum, x, y, and difference.
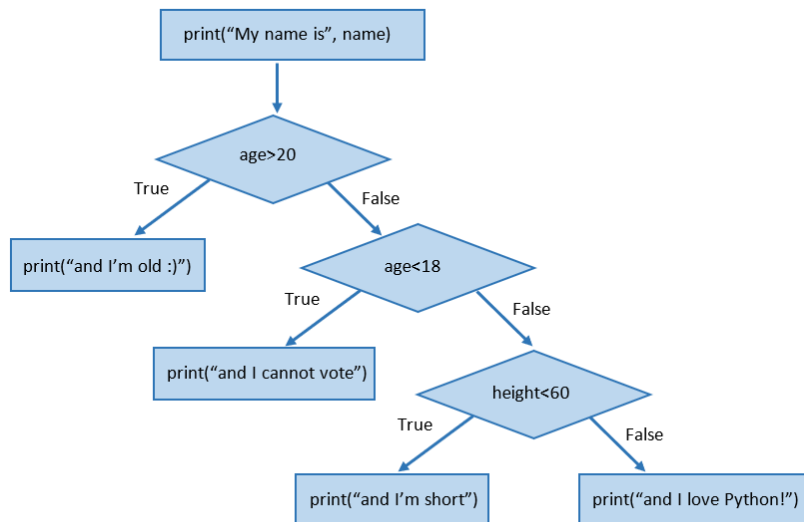
Booleans, Conditionals, and Errors

1. What will the following code output?

```python
def f(x, y, z):
    result = ""
    if (x + y) % 2 == 0:
        result += str(x)
    if (y + z) % 2 == 1:
        result = str(y) + result
    if z % 4 == 3:
        result = ""
    return result

print(f(1, -7, 526), f(8, 43, 2), f(9, 101, 11))
```

2. Write a function `canEatIceCream(temp, hunger)` to determine whether somebody should eat ice cream on a hot day based on the integer temp (must be greater than 60 degrees) and the float hunger (must be greater than 0.5). Return the result.

3. What is the difference between the and vs. or operations in terms of their relationship with the boolean `True`?

4. Convert the following flow chart into equivalent Python code.

```
                    print("My name is", name)
                              |
                              v
                          < age>20 >
              True    /              \    False
                     v                v
        print("and I'm old :)")    < age<18 >
                              True /         \ False
                                  v           v
                    print("and I cannot vote")   < height<60 >
                                          True /           \ False
                                              v             v
                              print("and I'm short")   print("and I love Python!")
```

5. Explain the following error messages and provide one possible fix:
   1)

```
Traceback (most recent call last):
  File "<tmp 1>", line 1, in <module>
    print("hello"+1)
TypeError: can only concatenate str (not "int") to str
```

2)

```
    print("name"",name)
                  ^
SyntaxError: EOL while scanning string literal
```

3)

```
def f():
    print("hello 15-110")
        return 15-110
```

```
  File "<tmp 1>", line 3
    return 15-110
    ^
IndentationError: unexpected indent
```

6. The following function takes in two numbers a and b, prints them as "a, b" and then returns their sum (for example, given a=3 and b=7 the function will print "3, 7" and it will return 10). Find the errors and classify them (into one of the three broad types of errors).

```
def printValuesReturnSum(a, b)
    print(a, b)
    return "a" +  b
```

7. The following lines of code contain multiple errors.

```
numerator = 0                    # line 1
denominator = 6                  # line 2
x = denominator/Numerator        # line 3
print("x', x)                    # line 4
```

1) Find the errors and categorize them into the three general types of errors discussed in class.
2) If you run these lines of code in Pyzo, it will produce an error message. State which error in the code will cause the error message and explain your answer. In other words, which error is detected first and why? (You should be able to answer this without actually running the code)

Circuits and Gates

1. Given the following boolean expression, fill out a truth table that shows all the possible results of the expression, then label the gates on the circuit below with AND/OR/etc. so that it produces the same results.

<p align="center">(A or B) and ((not B) xor C)</p>

Circuit:
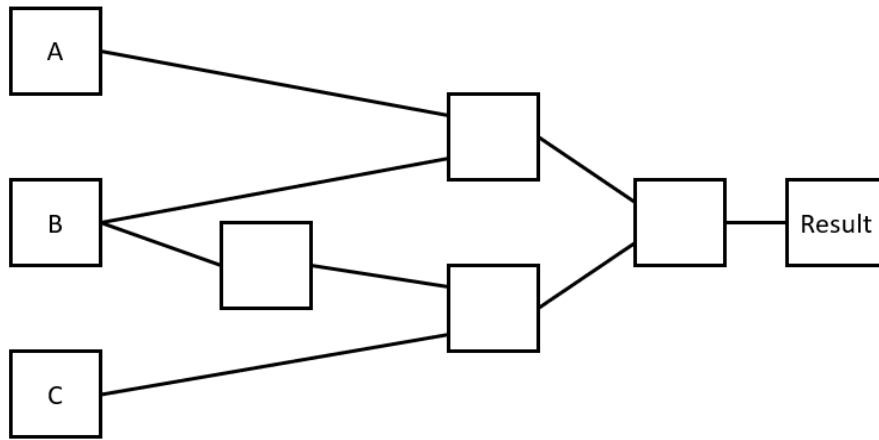
2. Recall that in lecture we built a simple addition machine called a Full Adder. Clearly name and describe the purpose of the input(s) and output(s) of this machine.

3. What is the main difference between a half adder and a full adder?

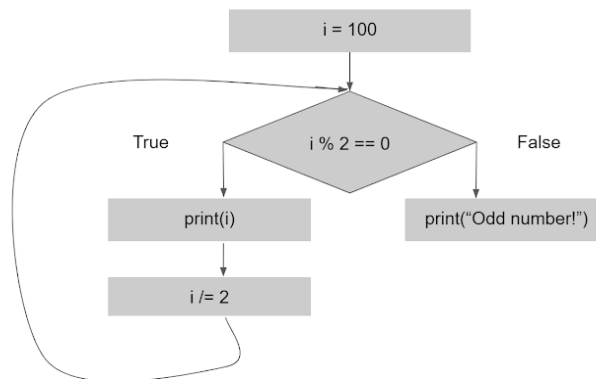4. What boolean operation does the following logic circuit behave like?
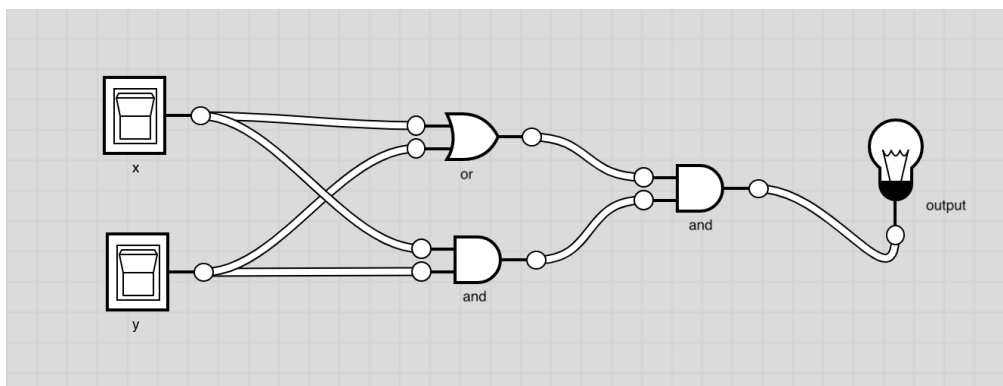
While Loops

1. Write a function cw1(n) that is algorithmically identical to the control flow chart shown below. The function should take an integer n as a parameter and **print** output as specified, returning nothing.
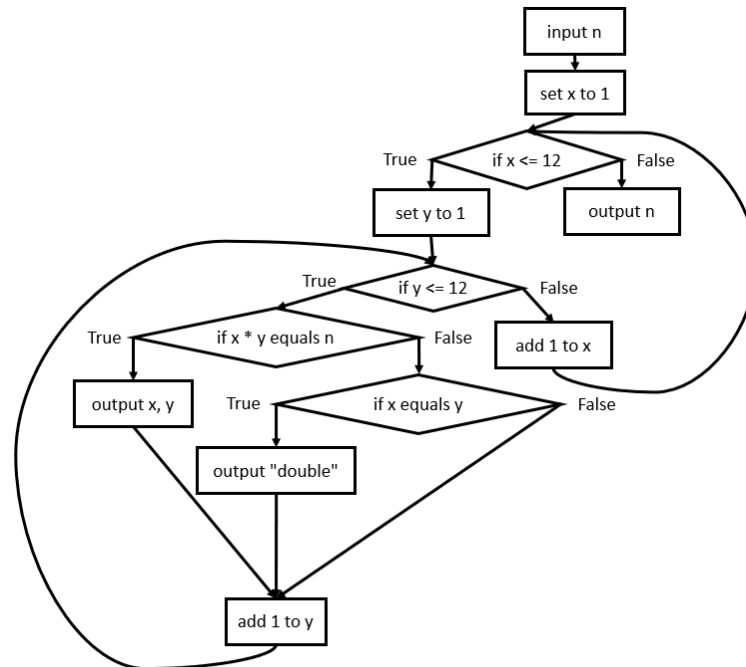
2. Write the while loop that corresponds with this flow chart.

3. Use while loop to write the function hasConsecutiveDigits(n) that takes in a possibly-negative int value n and returns True if that number contains two consecutive digits that are the same, and False otherwise.

4. Write the function isPowerOfFour(n) that takes in a number n and returns True if n is a power of 4, and returns False otherwise.

For Loops

1. Explain when you would use a while loop versus a for loop. Can you always convert a for loop to a while loop? Can you always convert a while loop to a for loop?

2. Write a function `numberOfFactors(n)` which takes in a positive integer and returns the number of factors it has.

3. Using a for loop, write the function `fizzBuzz(n)` that prints every number from 0 to n-1 inclusive. If the number is divisible by 3, print "fizz" instead of the number. If the number is divisible by 5, print "Buzz" instead of the number. If divisible by both 3 and 5, print "fizzBuzz" instead of the number.

4. Using a for loop, write the function `sumAllEven(n)` that finds the sum of all even numbers less than or equal to n.

Strings

1. Read through the following block of code, and write what it will output..

```
s = "Computer Science"
t = "GO-1-TEN"

print("A:", s[4])
print("B:", t[len(t)-2])
print("C:", s[6:12])

for i in range(2, 10, 4):
    print(s[i] + t[i])
```

2. Write a function whileSmile(s) that takes a string as input and uses a **while loop** to count the number of times the two-character string ":)" occurs in s. You should return the count. For example: whileSmile("Hello :):):)") should return 3.

3. Write a function reverseString(s) that returns a reversed version of the string s