

Name: _____ Andrew Id: _____

15-112 Spring 2024 Quiz 2

Up to 25 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use strings, loops, lists, tuples, dictionaries, try/except, or recursion on this quiz.

1. **Code Tracing:** Indicate what the following two programs print. Place your answers (and nothing else) in the boxes below to the code.

(a) (4 points) CT1

```
def f(x):  
    print(x, end=" ")  
    x = x // 2  
    print(x)  
    return x % 5  
  
x = 5  
a = 7  
print("x =", x, "; a =", a)  
print(f(a))  
print("x =", x, "; a =", a)
```

(b) (3 points) CT2

```
def f(x,y):  
    if x % y == 3:  
        y = x // 2  
        print("A1", x, y)  
    elif x % 2 == 0:  
        y = x * 2  
        print("A2", x, y)  
    if y == 8:  
        x = 4 * y + 2  
        print("A3", x, y)  
    return (x+y)//10 % 10  
  
def ct2():  
    print("L1", f(4, 7))
```

ct2()

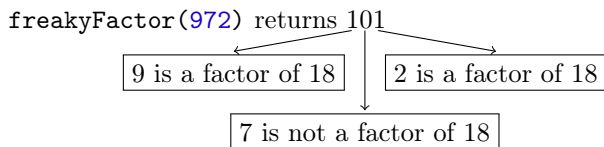
2. (6 points) **Free Response:**

Write a function `getTrianglePerimeter(x1, y1, x2, y2, x3, y3)` which calculates and returns the perimeter of a triangle given its coordinates. *Hint: It might be useful to use a helper function to find side lengths.*

3. (7 points) **Free Response:** Freaky Factors

Write the function `freakyFactor(n)` which, given a three digit number `n`, returns another three digit number that represents which digits in `n` are a factor of the sum of the digits in `n`. The returned number places a 1 in any locations where the corresponding digit of `n` is a factor of the sum, and 0 in any locations where the corresponding digit of `n` is not a factor of the sum.

Consider an example. If `n` is 972, then the sum of all the digits is $9 + 7 + 2 = 18$. The digit in the hundreds place, 9, is a factor of 18. The digit in the tens place, 7, is not a factor of 18. The digit in the ones place, 2, is a factor of 18. This means that return value will be 101, as follows:



Consider the following additional test cases:

```
# Normal cases
assert(freakyFactor(392) == 1)      # 2 is a factor of 14 (3+9+2)
assert(freakyFactor(323) == 10)   # 2 is a factor of 8 (3+2+3)
assert(freakyFactor(324) == 100)  # 3 is a factor of 9
assert(freakyFactor(134) == 101)  # 1 and 4 are factors of 8
assert(freakyFactor(413) == 110)  # 4 and 1 are factors of 8
assert(freakyFactor(426) == 111)  # 4, 2, and 6 are factors of 12
assert(freakyFactor(685) == 0)    # None of the digits are factors of 19

# Abnormal cases
assert(freakyFactor(99) == None)   # Not enough digits
assert(freakyFactor(1000) == None) # Too many digits
assert(freakyFactor(700) == 100)  # This works and does not crash
```