

Name: _____ Andrew Id: _____

15-112 Spring 2023 Quiz 2

Up to 20 minutes. No calculators, no notes, no books, no computers. Show your work!
Do not use strings, loops, lists, 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 next to the code.

(a) (6 points) CT1

```
# foo(n) is used by ct1
def foo(n):
    n + 1
    print("ok:", n // (n - 2))
    return n - 1

def ct1(x):
    y=foo(x%2 + foo(4))
    print("y:",y)

print("hello")
ct1(5)
```

(b) (6 points) CT2

```
# h(x,y) is used by ct2
```

```
def h(x,y):  
    print(x)  
    if (y > x):  
        print(y)  
        if (x < 5):  
            return 2  
        elif (x < 0 ):  
            return 9  
        print("112")  
    else:  
        return 6  
    return x  
  
def ct2(x):  
    res = h(1, x)  
    print("first =", res)  
    res = h(x, 0)  
    print("second =", res)  
  
print(ct2(-9))
```



2. (8 points) **Free Response:**

We will say that a value n is a **small 42ish** (coined term) if it is an integer number with precisely four digits and has at least one pair of consecutive digits that form the number 42.

For example, **4212**, **5042**, and **-4242** are **small 42ish** numbers because they are integers, have four digits, and **42** appears inside each of these numbers.

With this in mind, and **without using strings or loops**, write the function `isSmall42ish(n)` that takes a value n , which may or may not be an integer, and returns **True** if n is a **small 42ish** number, and **False** otherwise. Do not crash if n is not an integer! **Do not use strings or loops here.** Here are some test cases:

```
assert(isSmall42ish(121212) == False)    # more than 4 digits
assert(isSmall42ish(1542) == True)      # 4 digits, and 42 in the last two digits
assert(isSmall42ish(-4242) == True)     # 4 digits, and 42 in the first and last two digits
assert(isSmall42ish(1234) == False)     # missing 42 inside the number
assert(isSmall42ish(1) == False)        # only one digit
assert(isSmall42ish(42.42) == False)    # not an integer
assert(isSmall42ish("forty two") == False) # not an integer
```