

Name: _____ Andrew Id: _____

15-112 Fall 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. (7 points) **Code Tracing:** Indicate what the following program prints. Place your answer (and nothing else) in the box next to the code.

```
# foo(x) is used by ct1
def foo(x):
    x + 1
    return x - 1

def ct1(x):
    print(x + 3)
    y=foo(x%2 + foo(12))
    print("y:",y)

x=8
print("study")
print(ct1(x))
```


2. (3 points) **Reasoning Over Code:** Find an argument, `n`, for the following function to cause it to return `True`. Place your answer (and nothing else) in the box below the code.

```
import math
def roc(n):
    if abs(n) != n:
        return False
    else:
        return n**2 < 100 and math.isclose(round(n*100)%100, 42.0)
```


3. (10 points) **Free Response:**

We will say that a value n is a **small tidy** (coined term) if it is an integer number with precisely three digits and its digits are either strictly increasing or strictly decreasing in order.

For example, 123, 321, and -941 are **small tidy** numbers because they have three digits and their digits are in strictly increasing order (123) or strictly decreasing order (321 and -941). With this in mind, and without using strings or loops, write the function `isSmallTidy(n)` that takes a value n , which may or may not be an integer, and returns `True` if n is a **small tidy** 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(isSmallTidy(123))
assert(isSmallTidy(-123))
assert(isSmallTidy(961))
assert(isSmallTidy(111) == False)      # Not strictly increasing or decreasing
assert(isSmallTidy(1359) == False)     # More than 3 digits
assert(isSmallTidy(42) == False)        # Less than 3 digits
assert(isSmallTidy(54321) == False)    # More than 3 digits
assert(isSmallTidy("123") == False)    # Not an integer
assert(isSmallTidy(112) == False)      # Not strictly increasing or decreasing
assert(isSmallTidy(-963))
assert(isSmallTidy(-963.0) == False)   # Not an integer
```

