

# 15-110 Recitation Week 3

## Reminders

- Check 2 due Monday 2/06 at Noon EDT!
- Check your email if a TA can't find you during OH
- HW 1 grades are out!
- Check 1 and HW1 Revisions (+Ex revisions) due 2/07 at Noon EDT
- For recitation today:
  - There is a starter code file on the website
  - Recitation feedback form: <https://forms.gle/aUxeUEKpmdG8ZFNNZ>

## Overview

- Debugging
- Timed Function Practice
- Circuits and Gates

# Problems

## DEBUGGING

1. Rubber Duck Debugging



2. Printing and Experimenting

3. Thorough Tracing

### Catch the Error:

These are also in the starter file, with some additional code to help you.

For each function, specify **the type of error** (syntax/runtime/logical), **error name**, and **what the error means**.

```
1) def computeIntAverage(x, y, z):  
    sum = x + y + z  
    count = 3  
    return sum // count  
print(computeIntAverage(5, "6", 7))
```

|                |  |
|----------------|--|
| Type:          |  |
| Name:          |  |
| What it means: |  |

```
2) # leaving is a boolean that tells us if we should say hello or goodbye
def sayHelloOrGoodbye(leaving, name):
    if leaving == True:
        return "Hello "+ name
    else:
        return "Goodbye, "+ name+ ", and have a great day!"
```

|                       |  |
|-----------------------|--|
| <b>Type:</b>          |  |
| <b>Name:</b>          |  |
| <b>What it means:</b> |  |

3) import math

```
def circleArea(x0, y0, x1, y1):
    minRad = x_1 - x0 / 2
    maxRad = y1 - y0 / 2
    area = math.pi * minRad * maxRad
    return area
```

|                       |  |
|-----------------------|--|
| <b>Type:</b>          |  |
| <b>Name:</b>          |  |
| <b>What it means:</b> |  |

## TIMED FUNCTION PRACTICE

To help prepare for the exam, we will use this as a practice for writing code with a time limit. You will have 5 minutes to write the code independently (your TAs will help time this for you). Take out a piece of paper and pen (not in Thonny) and write the code in paper, to simulate an exam environment.

You're a professor trying to get through exam grading during finals week. Unfortunately, you're the only instructor and there are just too many exams to grade. You realize that you can create a function that decides each student's grade based on how many pages their exam is and the last number of their student ID.

You end up with this grading scale (the student will fail if they turn in 0 pages):

| Number of Pages | Last Digit of Student ID | Grade |
|-----------------|--------------------------|-------|
| More than 4     | Even                     | A     |
| More than 4     | Odd                      | B     |
| 4 and under     | Even                     | C     |
| 4 and under     | Odd                      | D     |
| 0               | -                        | F     |

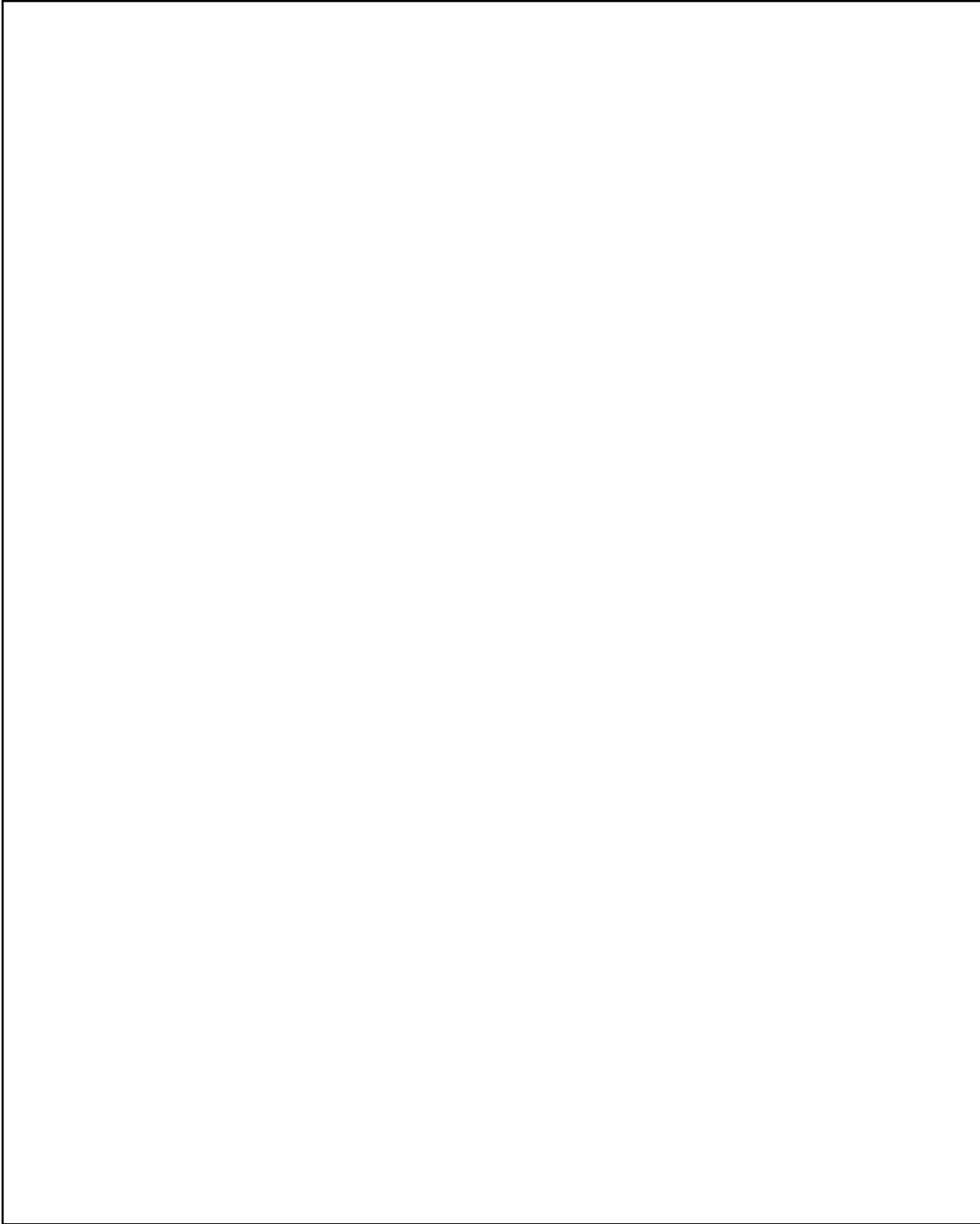
Write the function `studentGrade(numPage, studentID)` that takes in the number of pages that the exam is and the student's full ID number and implement the above grading scale.

**Note: You should nest if statements within the function and return early when necessary.**

Ex: `studentGrade(2, 123456) = "C"`

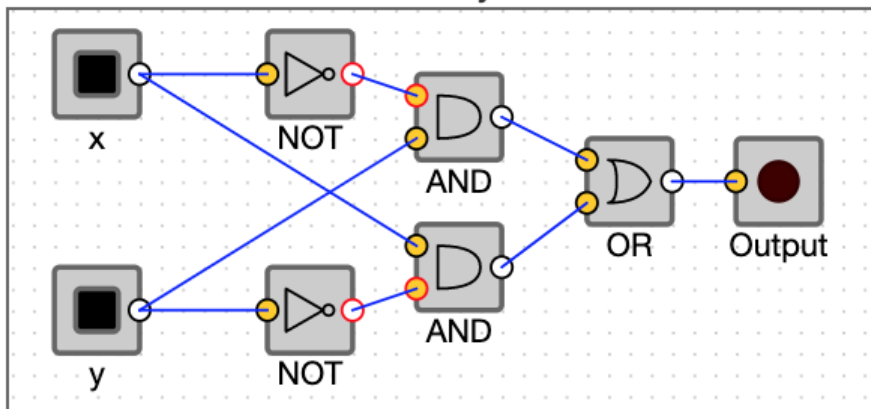
`studentGrade(5, 111112) = "A"`

*...write your code on the next page*



## CIRCUITS AND GATES

Write the equivalent Boolean expression demonstrated by the circuit:



$(\text{not } x \text{ and } y) \text{ or } (\text{not } y \text{ and } x)$

Fill out the truth table that corresponds to the circuit and boolean expression above:

| x | y | output |
|---|---|--------|
|   |   |        |
|   |   |        |
|   |   |        |
|   |   |        |

Logic.ly