**Alvaro Salinas**

**Project Plan for Assignment 3a**

Testing Plan:

**As per assignment instructions, we can assume users will always enter at least 1 integer. Therefore, not creating data validation for this input and not including in testing plan

| How many integers would you like to enter? | Integers entered by user | Output |
|---|---|---|
| 1 | 0 | min: 0; max: 0 |
| 4 | -4; 105; 2; -7 | min: -7; max: 105 |
| 5 | -4; -3; -2; -1; 0 | min: -4; max: 0 |
| 5 | 0; 1; 2; 3; 4 | min: 0; max: 4 |
| 2 | 999,999; -999,000 | min: -999,999; max: 999,999 |
| 3 | -5; -5; -5 | min: -5; max: -5 |
| 3 | 5; 5; 5 | min: 5; max: 5 |

Design:

Initialize *minimum integer* value to 0
Initialize *maximum integer* value to 0

Ask user how many integers they would like to enter
*[User enters number of integers]*

Ask user to enter integers (as many as they previously chose)
*[User enters integers]*

Initialize *integer counter* value to 1

While *integer counter* value is less than or equal to the number of integers the user decided to input:

> If integer entered by user is greater than current *maximum integer* value, set that integer as new *maximum integer* value
>
> Else if integer entered by user is less than current *minimum integer* value, set that integer as new *minimum integer* value
>
> Else, the minimum and maximum integer values do not get updated
>
> Add 1 to the integer counter
>
> [Previous four steps get applied to every integer that the user entered]

Display the final *minimum integer* and *maximum integer* values. [These are the minimum and maximum values from the set of integers entered by the user.]