# Software Project I: Algorithms

## Purpose

To create (a) product(s) that engage(s) you and that you would be proud to share to a public audience.

Along the way, you will develop your ability to problem-solve using a variety of strategies, to design an algorithm, to implement a solution in code, to manage source code using accepted industry practices, and to plan and meet commitments for project milestones.

## Evaluation

75% of your final grade on a product comes from your ability to provide regular evidence that you have met curriculum expectations in this course.

Using your final commit GitHub, and your posts on Sesame, what have you learned and demonstrated knowledge of?

You probably will not have demonstrated all of the expectations listed, but have you hit the majority of the expectations?

How well? Did you develop the ability to meet these expectations independently? Did you challenge yourself?

Add more example

## Curriculum Expectations I Believe I Have Met

### A1. Data Types and Expressions Demonstrate the ability to use different data types, including one-dimensional arrays, in computer programs;

**A1.1** use constants and variables, including integers, floating points, strings, and Boolean values, correctly in computer programs;

ASCII, Unicode) to internally represent data and store information;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L17-L22](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L17-L22)

Here are some examples of the variable type I used

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L89](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L89)

Later I use integers in the median algorithm.

**A1.3** use assignment statements correctly with both arithmetic and string expressions in computer programs;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L16-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L16-L47)

Here I use the Boolean instead of integers to control this while loop and make it more efficient. As well thoughout the rest of the code there are many examples.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L147-L160](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L147-L160)

**A1.4** demonstrate the ability to use Boolean operators (e.g., AND, OR, NOT), comparison operators (i.e., equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to), arithmetic operators (e.g., addition, subtraction, multiplication, division, exponentiation, parentheses), and order of operations correctly in computer programs;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L95-L103](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L95-L103)

Here I use Mod operator to figure out weather or not the int is even or odd.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L123-L139](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L123-L139)

Here I shows many example of using comparative operators.

**A1.5** describe the structure of one-dimensional arrays and related concepts, including elements, indexes, and bounds;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L22](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L22)

Here is the main data storage of my program I use arrays to perform the calculation on.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L116-L119](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L116-L119)

As well I use dictionaries in my code to hold data.

**A1.6** write programs that declare, initialize, modify, and access one-dimensional arrays.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L68-L72](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L68-L72)

Throughout my program I use access the main array many times here is one example of it.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L123-L139](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L123-L139)

I use dictionaries as well to get my program to count the amount of certain types of numbers there are in an array.

### A2. Data Types and Expressions Demonstrate the ability to use control structures and simple algorithms in computer programs;

**A2.1** write programs that incorporate user input, processing, and screen output;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L24-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L24-L47)

This loop incorporates my program user input and visual input. It takes in a number and puts it into an array.

**A2.2** use sequence, selection, and repetition control structures to create programming solutions;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L123-L139](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L123-L139)

This counts the amount of times a number appears in the array.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L147-L160](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L147-L160)

I used this algorithm to put perform the standard deviation formula in my program.

**A2.3** write algorithms with nested structures (e.g., to count elements in an array, calculate a total, find highest or lowest value, or perform a linear search).

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L133-L139](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L133-L139)

This shows a algorithm to find the highest value in the array.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L59-L104](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L59-L104)

In these two examples I was able to calculate the mean and the meadian using algorithms.

### A3. Subprograms Demonstrate the ability to use subprograms within computer programs;

**A3.1** demonstrate the ability to use existing sub-programs (e.g., random number generator, substring, absolute value) within computer programs;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L85](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L85)

Used the sorting function in swift.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L21](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L21)

Here I use the read line function in my code

**A3.2** write subprograms (e.g., functions, procedures) that use parameter passing and appropriate variable scope (e.g., local, global), to perform tasks within programs.

I don’t have any functions in my code.

### A4. Code Maintenance Use proper code maintenance techniques and conventions when creating computer programs.

**A4.1** demonstrate the ability to identify and correct syntax, logic, and run-time errors in computer programs;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L17-L22](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L17-L22)

I used the right syntax for converting variables.

**A4.2** use workplace and professional conventions (e.g., naming, indenting, commenting) correctly to write programs and internal documentation;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L81-L103](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L81-L103)

I used appropriate variables names in the code.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L95-L103](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L95-L103)

This sections of code shows understanding of syntax and the use of proper brackets.

**A4.3** demonstrate the ability to interpret error messages displayed by programming tools (e.g., compiler, debugging tool), at different times during the software development process (e.g., writing, compilation, testing);

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L123-L165](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L123-L165)

I was able to clear up all the syntax errors using the compiler in my program.

**A4.4** use a tracing technique to understand program flow and to identify and correct logic and run-time errors in computer programs;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L162-L165](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L162-L165)

I used the print function to debug my program and make sure it was doing what I wanted it to do.

**A4.5** demonstrate the ability to validate a program using a full range of test cases.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L26-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L26-L47)

In my program It checks to make sure that the input is valid and then allows the user to insert another number.

### B1. Problem-solving Strategies Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team;

**B1.1** use various problem-solving strategies (e.g., stepwise refinement, divide and conquer, working backwards, examples, extreme cases, tables and charts, trial and error) when solving different types of problems;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L26-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L26-L47)

I used a couple of these strategies when making this part of the code I used a lot of guess and check, as well I used stepwise refinement to make sure the code was doing the things it should.

**B1.2** demonstrate the ability to solve problems independently and as part of a team;

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L26-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L26-L47)

For this part of the code me and Tommy Davies worked together to get it done and it ended up being very effective as we got the algorithm done relatively quickly.

**B1.3** use the input-process-output model to solve problems.

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L26-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L26-L47)

I used this in my whole product as the user puts there input into the program and gets a summarized version of there data.

### B3. Designing Algorithms Use a variety of problem-solving strategies to solve different types of problems independently and as part of a team;

**B3.1** design simple algorithms according to specifications.

<https://raw.githubusercontent.com/rsgc-elder-a/-algorithms-isp/master/ipo.txt>

[https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics ISP/main.swift - L21-L47](https://github.com/rsgc-elder-a/-algorithms-isp/blob/master/Statistics%20ISP/main.swift#L21-L47)

I wrote a algorithm first on this text file and then implemented it into my code this shows I can design simple algorithms.

## Final Comments and Proposal for Level of Achievement

Taking into consideration the purpose of this project and the evaluation criteria, what overall level of achievement do you feel you have earned?

I fell I have earned a mark somewhere around a 90% I have shown proof that I was able to complete all these challenge at least once and more then once in many cases.