ASSESSMENT AND INTERNAL VERIFICATION FRONT SHEET (Grouped Criteria)

(Note: This version is to be used for an assignment brief issued to students via Classter)

Course Title	**BSc. Software Development** **BSc. Creative Computing** **BSc. Applied Data Science** **BSc. Digital Games Development**	**Lecturer Name & Surname**	**Andrew Cortis** **Kassandra Calleja**		
Unit Number & Title	ITSFT-506-1608 – Data Structures & Algorithms				
Assignment Number, Title / Type	2 – Advanced Algorithms Implementation & Evaluation / Home				
Date Set	12/05/2025	**Deadline Date**	09/06/2025		
Student Name	Gerrard Compton	**ID Number**	306506L	**Class / Group**	SWD 6.1A

Assessment Criteria	Maximum Mark
KU2.7 : Explain the process known as the Fisher Yates Shuffle. *KU3.1: Show analysis of estimate running times and compare implementation of efficient algorithms with inefficient algorithms.* *AA2.3: Produce an algorithm for Graphs or Tree structures. Also compute the best, worst and average case times.* *AA2.4: Produce an algorithm using the Binary Tree structure. Also compute the best, worst and average case times.* *AA2.5: Produce an algorithm using the queue data structure to prioritize data. Use a data structure such as a Heap and the Heapsort algorithm.* *SE2.6: Evaluate the applications of pseudo random number generator.* *SE2.8 : Implement three different sorting algorithms. Predict the rate of processing and evaluate and justify application for each algorithm.* *SE4.1 : Evaluate the features algorithms in relation to their correctness, proof and intractability.* ***Total Mark***	61

Notes to Students:

- This assignment brief has been approved and released by the Internal Verifier through Classter.
- Assessment marks and feedback by the lecturer will be available online via Classter (http://mcast.classter.com) following release by the Internal Verifier
- Students submitting their assignment on Moodle/Turnitin will be requested to confirm online the following statements:

Student's declaration prior to handing-in of assignment

❖ I certify that the work submitted for this assignment is my own and that I have read and understood the respective Plagiarism Policy

Student's declaration on assessment special arrangements

❖ I certify that adequate support was given to me during the assignment through the Institute and/or the Inclusive Education Unit.

❖ I declare that I refused the special support offered by the Institute.

MCAST

ITSFT-506-1608
Data Structures & Algorithms

First Year BSc. 2024-2025

Assignment 1

Advanced Algorithms Implementation & Evaluation

Assignment Guidelines

Read the following instructions carefully before you start the assignment.
If you do not understand any of them, ask your lecturer.

- This is a **HOME** Assignment to be completed by the **DEADLINE SPECIFIED BY LECTURER ON VLE**.
- The assignment consists of **2 Sections and CARRIES 61marks**; all tasks must be attempted.
- Please note that **ALL WORK** must be handed in by the stipulated deadlines. **LATE ASSIGNMENTS WILL NOT BE ACCEPTED.**
- The assignment sheet and assignment coversheet should be fully completed with all the necessary details. Note that **assignments handed in without the assignment cover sheet are considered as not submitted.**
- Assignments ***must* be handed in as a soft copy uploaded on Moodle** by the stipulated deadline.
- Any **references should be listed and quotes should be paraphrased *properly***. Unless listed and paraphrased properly the assignment will be regarded as plagiarized. **Referencing should be carried out using IEEE Style** Referencing Notation.
- **Copying is strictly prohibited and will be penalized** in line with the College's disciplinary procedures.

Task 2.2 – PRNG Intractability & Randomness

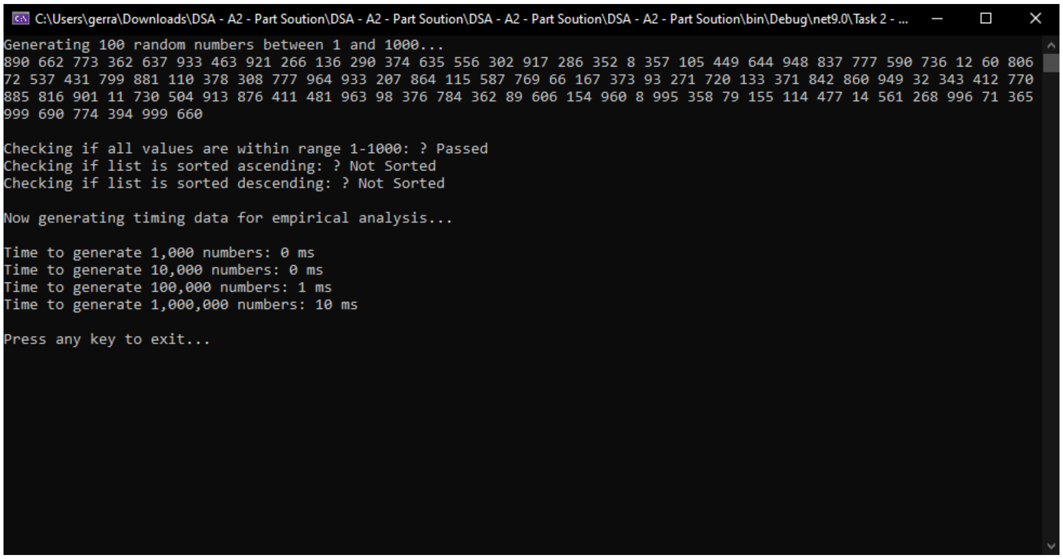
To evaluate the quality and performance of the SplitMix64 PRNG implementation, 100 random numbers were generated within the range of 1 to 1000. The following criteria were assessed:

- All numbers fall within the specified range ✅

- The list is neither sorted ascending nor descending ✅

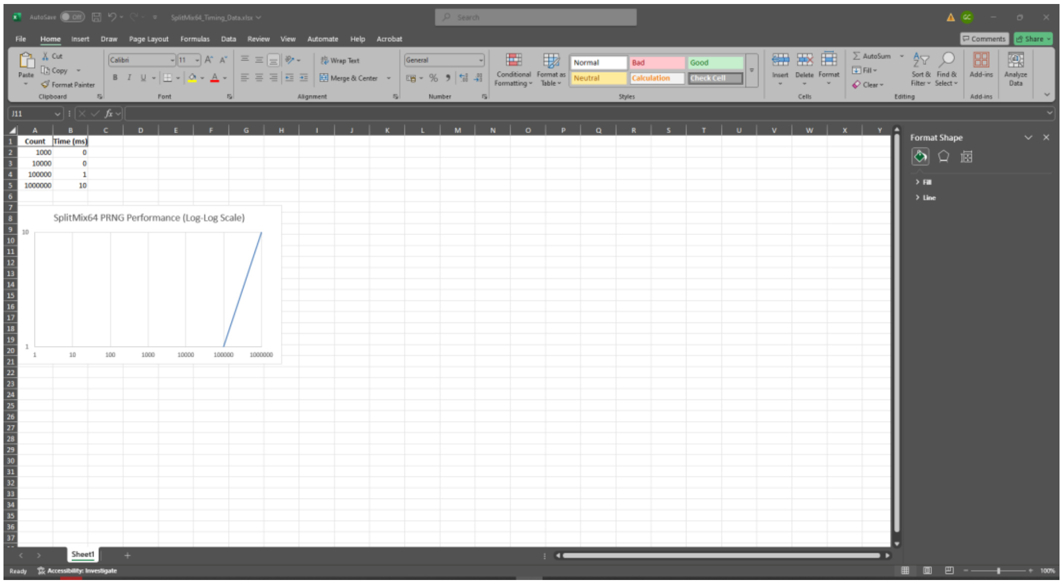
- Time was measured for generating 1K, 10K, 100K, and 1M values

Below is a screenshot from the output of the program:


C:\Users\gerra\Downloads\DSA - A2 - Part Soution\DSA - A2 - Part Soution\DSA - A2 - Part Soution\DSA - A2 - Part Soution\bin\Debug\net9.0\Task 2 - ...

```
Generating 100 random numbers between 1 and 1000...
890 662 773 362 637 933 463 921 266 136 290 374 635 556 302 917 286 352 8 357 105 449 644 948 837 777 590 736 12 60 806
72 537 431 799 881 110 378 308 777 964 933 207 864 115 587 769 66 167 373 93 271 720 133 371 842 860 949 32 343 412 770
885 816 901 11 730 504 913 876 411 481 963 98 376 784 362 89 606 154 960 8 995 358 79 155 114 477 14 561 268 996 71 365
999 690 774 394 999 660

Checking if all values are within range 1-1000: ? Passed
Checking if list is sorted ascending: ? Not Sorted
Checking if list is sorted descending: ? Not Sorted

Now generating timing data for empirical analysis...

Time to generate 1,000 numbers: 0 ms
Time to generate 10,000 numbers: 0 ms
Time to generate 100,000 numbers: 1 ms
Time to generate 1,000,000 numbers: 10 ms

Press any key to exit...
```

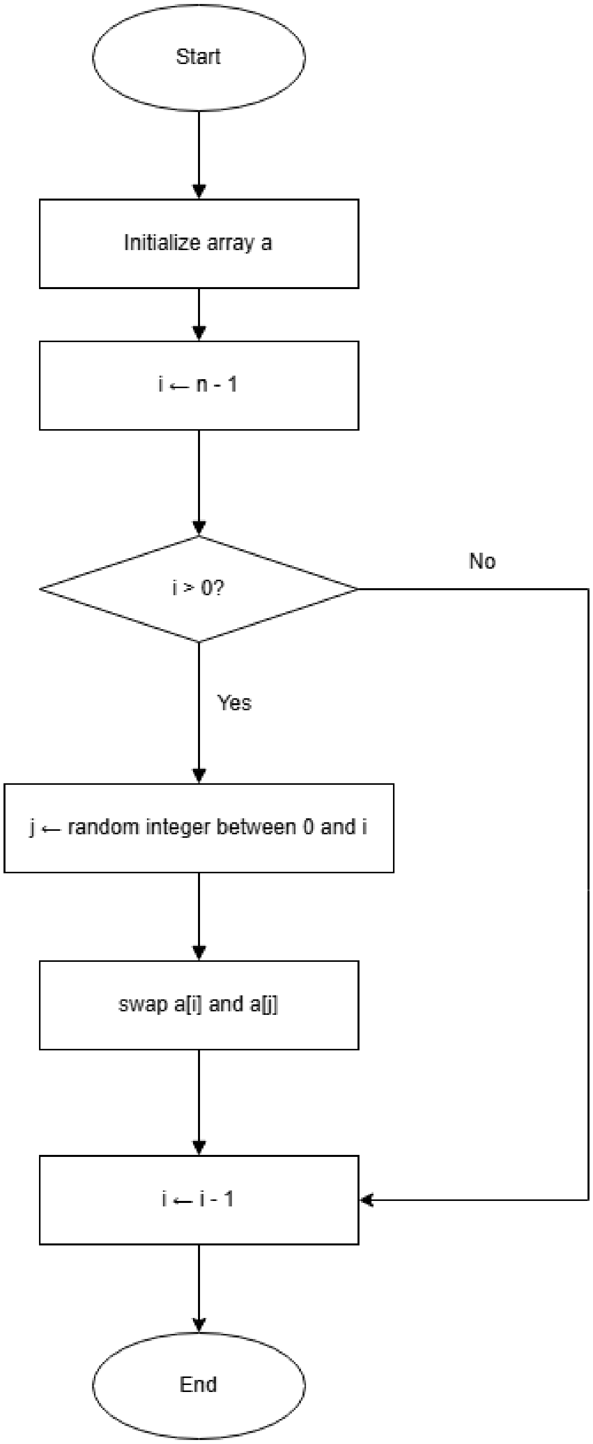
The empirical results were plotted on a log-log graph:



Count	Time (ms)
1000	0
10000	0
100000	1
1000000	10

Task 2.3 – Fisher-Yates Shuffle Flowchart

The following flowchart illustrates the Fisher-Yates Shuffle algorithm, which randomly reorders elements in an array using SplitMix64 as the random number generator.



This algorithm ensures uniform randomness by swapping each element from the end of the array with a randomly selected index from the remaining unshuffled portion.