

Math for computing final

Student Name: Raneem Yahya Sadeh

Student ID: 22210016

Instructor Name: Drs. Bayan Shabanah

Course Name: Math For Computing

HTU Course No: 40303121

BTEC Unit No: Unit 14(R/618/7421)

Academic Year Summer 2023

A grey logo on a black background

Description automatically generated with low confidence

**Math For Computing**

**Contents**

[**What is Math For Computing?** 4](#_Toc144463251)

[**Task 1:** 4](#_Toc144463252)

[**Step A:** 5](#_Toc144463253)

[**Step B:** 6](#_Toc144463254)

[**Task 2:** 6](#_Toc144463255)

[**The results:** 7](#_Toc144463256)

[**Task 3:** 9](#_Toc144463257)

[**The coordinates for the shape:** 9](#_Toc144463258)

[**The robot shape output:** 10](#_Toc144463259)

[**References:** 11](#_Toc144463260)

# **What is Math For Computing?**

It refers to the branch of mathematics that is particularly relevant and useful in the context of computer science and related fields. Computer science heavily relies on mathematical concepts and techniques to solve problems, analyze algorithms, design efficient data structures, and develop software. Math for Computing covers a wide range of topics, many of which are particularly pertinent to computer science, including:   
Probability and Statistics: Probability is vital in understanding uncertainty and randomness, prevalent in many computer science applications. Statistics is used in data analysis, machine learning, and experimental design.

Number Theory: Number theory plays a role in cryptography, as it involves the study of integers and their properties. It's used to create secure encryption algorithms.

Computation Theory: This includes automata theory, formal languages, and computability theory. These concepts are fundamental to understanding the limits and capabilities of computation.

Graph Theory: Graphs are used to model relationships between objects and are essential in fields like networking, social network analysis, and optimization problems.

Differential Equations: These are used to model dynamic systems in computer simulations and scientific computing.

Numerical Methods: Numerical methods involve approximating solutions to mathematical problems using computers. They are used in various computational applications.

Algebraic Structures: Concepts from abstract algebra, such as groups, rings, and fields, can be applied in coding theory, cryptography, and data compression.

# **Task 1:**

***Explain how prime numbers are important by evaluating an example based on the RSA algorithm.***

1. Using my provided University number **(22210016)**, let's dissect the RSA algorithm's phases. A common encryption technique that depends on the characteristics of prime numbers is the RSA algorithm. Using a public key for encryption, and a private key for decryption. Because it is challenging to factor huge composite numbers into their prime components, the algorithm's security is predicated on this fact.

**Here are the steps:**

**Step A: Generating a Public/Private Key Pair**

I already have **P = 23** and by taking the last two digits of my university number, which is 16. The closest single-digit prime is 17 **q = 17**.

To get (n) we 🡪 **p \* q = 23 \* 17 = 391**

To get φ(n) we 🡪 **(p - 1) \* (q - 1) = 22 \* 16 = 352**

**Then I will Choose a value for e (public exponent):**

I will choose a small prime number that is greater than 1 and less than φ(n) and is coprime to φ(n). Let's choose **e = 7**

Then I will find the modular multiplicative inverse of e which is (d):

(d \* e) % φ(n) = 1

First, I found:

φ(n) = 352, e = 7

352 = 50 \* 7 + 2

7 = 2 \* 3 + 1

2 = 2 \* 1 + 0

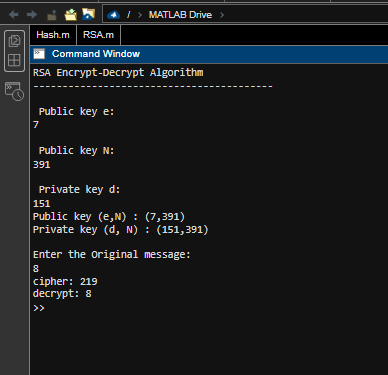
1 = 7 - 2 \* 3

= 7 - (352 - 50 \* 7) \* 3

= 151 \* 7 - 352 \* 3

151 \* 7 - 352 \* 3 ≡ 1 (mod 352)

This means that **d = 151**

**Step B:**

Encrypting and Decrypting

Using the code I solved step b and by the given message (M=8).

# **Task 2:**

***Evaluate probability theory to an example involving hashing and load balancing.***

I'm being asked to apply probability theory to a specific hashing and load-balancing scenario in this question. The example uses MATLAB code with a code name "Hash\_Summer2023" that defines two hash functions with various techniques. I must simulate, examine, and compare how client requests are distributed among servers for each hash function.

I have initialization:

n=80 : The number of servers.

request\_n=33000 : The number of client requests.

Hash Function Based on Modular Arithmetic, I will define parameters for the hash function:

p= 2617: A prime number.

a=50: A coefficient.

b=40: An offset.

**Hash Function Based on Truncation:**

To create a key (S) for this hash function, portions of the random integer (which represents the client's IPv6 address) must be converted. The key is next transformed into a numeric value and put through the modulo operation with the number of servers (n), and it is already in the code.

This method attempts to achieve uniform distribution, however, due to the nature of IPv6 addresses and the characters chosen for the key, it might not be fully uniform. The address's different trends in some areas may impact the distribution.

**Hash Function Based on Modular Arithmetic:**

The second hash function generates hash values via a mathematical technique employing modular arithmetic. When factors like the prime number p, coefficient a, and offset b are carefully selected, this strategy is intended to distribute values more evenly.

Due to a variety of factors, including the characteristics of the input data, the nature of the hash function, and the distribution of the generated random numbers, obtaining absolute uniformity in hash functions can be difficult in reality. While both approaches aim for uniform distribution, the degree of uniformity could differ based on the implementation specifics and parameter selections.

## **The results:**

**The results of the two hash functions are compared.**

A computer screen with green and white text

Description automatically generated

**The histograms for both hash functions are plotted side by side for visual comparison.**

**And as you see they are not uniform because they are not equal**

A graph of a number of columns

Description automatically generated with medium confidence

**The mean and standard deviation of both distributions are calculated and compared.**

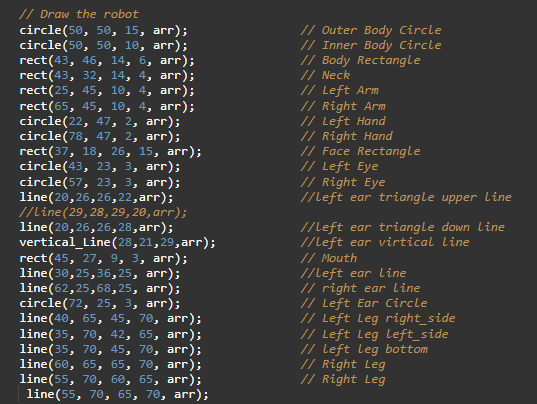
|  |  |
| --- | --- |
| **The distribution's central tendency and spread are better understood by looking at the mean and standard deviation figures. In general, it shows greater load-balancing performance if the mean is close to uniform and the standard deviation is low. We can determine which hash method produced a more evenly distributed server allocation by comparing the means and standard deviations of the two distributions.** | **A screenshot of a computer program  Description automatically generated** |

# **Task 3:**

***Evaluate the Co-ordinates system used in programming a simple output device.***

This application shows how geometric forms can be drawn in code and how coordinate systems work. It positions and draws each element of the robot's visual representation using a coordinate-based method. You can alter the coordinates to alter the sizes and locations of the various forms to produce various visual effects.

## **The coordinates for the shape:**



## **The robot shape output:**

****

# **References:**

Digital BitHub. (n.d.). Division Modulo Method: Hashing Technique. [Online]. Available at:

<https://www.digitalbithub.com/learn/division-modulo-method-hashing-technique>

Kelsey, J. (n.d.). Truncation and Multi-collision Attacks on the ISO/IEC 10118-3:2004 Cryptographic Hash Functions. [Online]. Available at:

<https://csrc.nist.gov/CSRC/media/Events/First-Cryptographic-Hash-Workshop/documents/Kelsey_Truncation.pdf>

Computer Science Degree Hub. (n.d.). 5 Types of Math Used in Computer Science. [Online]. Available at:

<https://www.computersciencedegreehub.com/lists/5-types-of-math-used-in-computer-science/>

Mathnasium. (n.d.). Five Types of Math Used in Computer Science. [Online]. Available at:

<https://www.mathnasium.com/bend/news/five-types-of-math-used-in-computer-science>