Modulo Calculator Using Tkinter Library

Kuldeep Vayadande, Samruddhi Pate, Naman Agarwal,
Dnyaneshwari Navale, Akhilesh Nawale and Piyush Parakh

March 17, 2022

MODULO CALCULATOR USING TKINTER LIBRARY

(Kuldeep Vayadande, Samruddhi Pate, Naman Agarwal, Dnyaneshwari Navale, Akhilesh Nawale, Piyush Parakh)

Artificial Intelligence and Data Science, Vishwakarma Institute of Technology, Pune, Maharashtra, India

kuldeep.vayadande1@vit.edu, samruddhi.pate20@vit.edu, naman.agarwal20@vit.edu, dnyaneshwari.navale20@vit.edu, akhilesh.nawale20@vit.edu, piyush.parakh20@vit.edu,

Abstract — **Modular arithmetic is a branch of mathematics that uses the "mod" feature. Let's imagine we're dividing two integers, A and B. When we divide A by B, we are sometimes simply interested in the leftover, i.e. the remainder. There is a modulo operator that can be used in these situations (abbreviated as mod). The computation of "mod" of expressions is the focus of modular arithmetic. Expressions can contain digits as well as addition, subtraction, multiplication, division, and other computational symbols. In this project, we are basically making a modulo calculator, which performs arithmetic operations modulo p over a given math expression. While you may still use this modulo calculator to determine the remainder of Euclidean division by a specific modulus by entering an integer value, it can do a lot more. You can also enter a math expression involving other integers as well as a variety of modular arithmetic operations like addition, multiplication, division, subtraction, exponentiation, and so on. All procedures will be performed with a modulus in mind. Clock arithmetic is another name for modular arithmetic. This is because, much as a clock resets to zero at midnight, the number resets itself each time the modulus, or mod, is reached, causing it to wrap around the modulus.**

Keywords *— Modular Arithmetic, Modulus, Arithmetic Mathematics, mod, remainder, modulo, clock arithmetic, modulo calculator, addition, subtraction, multiplication, division, square root, GCD, LCM, Primality, XOR, Factorization, module, exponentiation, discrete logarithm, module inverse*

I. INTRODUCTION

On the opening page of his magnum opus Arithmetical Investigations, also known as Disquisitions Arithmeticae, Carl Friedrich Gauss established modular arithmetic. Gauss was a talented mathematician who lived during the Renaissance. His brilliance was apparent from an early age. His primary school instructor assigned the class a task: in order to get some tranquilly, he ordered his students to add the first 100 numbers. Gauss, who was only eight years old at the time, spotted a pattern that led straight to the answer, whereas his colleagues went head-on into the difficulty. He matched one with one hundred, two with ninety-nine, and so on until he reached 50 and 51. The total must be 5,050 because each of the 50 pairings adds up to 101. Gauss completed Disquisitiones Arithmeticae while he was only 21 years old, and it was published three years later in 1801. This paper revolutionized number theory by combining numerous prior findings and including much that was new. It was one of the final Latin-written mathematical works. This book's logical framework set a precedent that was followed by many subsequent texts. Gauss' work influenced the theory of numbers in a variety of ways, and his influence lasted well into the twentieth century.

Modular arithmetic, also termed as clock arithmetic, is arithmetic conducted with a count that resets to zero every time the modulus (mod) reaches a certain whole integer N greater than one. Two examples are a 24-hour digital clock that resets to 0 at midnight (N = 24) and a 360-degree circle protractor (N = 360). In number theory, modular arithmetic is important because it is a key tool for addressing Diophantine problems Modular arithmetic is an integer based arithmetic method that takes the remainder into account. When a number reaches a specific fixed quantity (known as the modulus), it "wraps around" to leave a remnant. Wilson's theorem, Lucas' theorem, and Hensel's lemma are all examples of modular arithmetic, which is commonly used in domains such as cryptography, computer science, and computer algebra. The clock example used to demonstrate its use is a 12-hour clock using modular arithmetic. If it is 10:00 now, the clock will reflect 3:00 instead of 15:00 in 5 hours. With a modulus of 12, 3 is the remainder of 15. When a number x mod N is divided by N, it is similar to asking for the remainder of x. If two integers a and b have the same remainder when divided by N, they are said to be congruent (or in the same equivalence class). If this is the case, we say that ab (mod N).

Modular arithmetic is a cornerstone of number theory in pure mathematics and is widely employed. It does, however, have a wide range of applications. It's used to find problems in international standard book numbers (ISBNs) and bank IDs (Iban numbers) by calculating checksums. Public key encryption methods, which are critical for modern commerce, are also based on modular arithmetic. It's also popular in the field of computer science. Finally, modulo 12 arithmetic is used in music theory to analyze the 12-tone equal temperament system.

Various operations can be performed on it such as Addition, Division, Exponentiation, Factorization, GCD,LCM, and many more, and in our Project, many of them, to be precise 15 of them are implemented with a mod facility available in many. It appears to be a simple calculator, but it is more than that. It contains various operations. Also, the history of the previous session is saved, giving the user a better experience. This Report contains the overall description of the whole project and mentioning the concept of MODULAR ARITHMETIC. Firstly, it contains the history of the introduction of Modular Arithmetic. The real life applications it can be used are specified further in the introduction. Then some papers which we referred to and studied are briefly introduced in the Literature Review section. Then, in the methodology section, the core part of methodology and the process used for the project is discussed with a few flowcharts and diagrams. Then, comes the Result and Discussion section, which contains a few screenshots of our project's output. Then, is the final conclusion of the project with Future Scope, of where in future Modular Arithmetic can be advanced more and be used expensively, also how this project can be extended further. And at the end is the references part with few links of papers and study materials referred for this project.

II. LITERATURE REVIEW

Various studies suggest benefits of Modular Arithmetic and its Applications in daily life .Some of these studies are discussed here.

M. K. Stojčev[1] The paper proposes a unique approach for computing "mod" operations in high-level languages such as C, Java, C++, Mathematica, and Matlab that removes any ambiguities. Modular arithmetic allows us to do algebraic calculations on integers while systematically avoiding terms divisible by a specific number (called the modulus). Johann Großschädl [2] The project's main purpose was to figure out how to build fundamental arithmetic algorithms for public-key cryptography while using as little energy as feasible. Several algorithms have been devised, however they consume a lot of power, increasing the energy expenses of arithmetic methods..

XiaotianWu [3] The document that follows includes A method of secretly transmitting photos with higher visual quality. Alternative alteration strategies are provided and thoroughly studied; the best option is also determined.

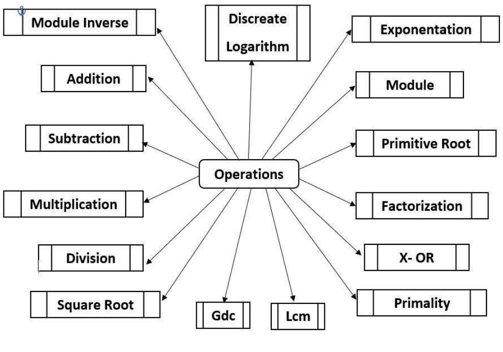
R. Chaves 1 and L. Sousa 1 [4] RNS are non-weighted systems allowing each residue to perform add, sub and multiply operations simultaneously and independently. Because the arithmetic units for individual channels and converters to and from these are simple.
Thomas Plantard [5] Modular multiplication is used in a number of applications. In general, large size moduli are the focus of contemporary modular multiplication approaches in the literature.

III. METHODOLOGY/EXPERIMENTAL

This project is basically a Modular Arithmetic Calculator that executes modulo p arithmetic operations on a given math phrase.
This is being implemented in **Python language** and **Tkinter library** is used for making of the GUI. Even **Math Module** is used for calculations.

Tkinter is Python's standard graphical user interface package. Python with Tkinter makes it simple to design graphical user interfaces. It gives the Tk GUI toolkit a rich object-oriented interface. It supports many controls in a GUI programme, such as buttons, labels, and text boxes. Widgets are the general name for these controls. **Math Module** -This can be used as a built-in module in Python to perform mathematical tasks.

This project is formulated in **VS Code IDE** and is implemented on Command line.
There is no such extra hardware requirement for implementation of this Project. In Software, you just need Python and the required modules installed in your system. While you may still use this modulo calculator to determine the remainder of Euclidean division by a specific modulus by entering an integer value, it can do a lot more.
You may also enter the math expression containing other integers and the many modular arithmetic operations like addition, multiplication, division, subtraction, exponentiation, etc. All operations will be carried out taking a modulus into account.
And the Operations we have added in our project are - (fig1) :



(fig -1)

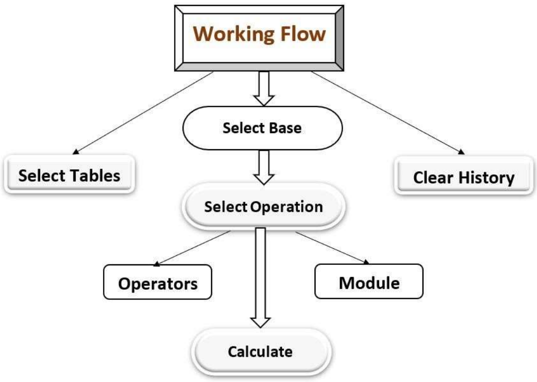
Here, the user has to follow simple steps to use this calculator -

1. Start the Calculator
2. Select **Base** (for now just base 10 is available) 3. Then select the **operations** you want to perform, be it addition, subtraction, finding of prime or getting module inverse.

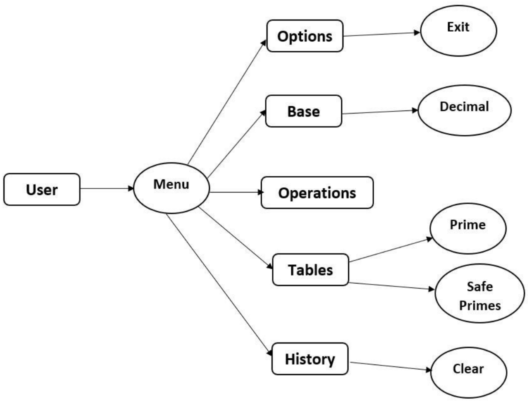
with this many more- in total 15 operations are available in the calculator to be performed.

4. Then enter the **values for the operators** and the **module value**, which is optional.
5. And then finally just press on the **Calculate** button.

The **History** of the same will be displayed on the **right side of the window** and can be cleared anytime. (fig-2)



(fig-2)



(fig-3)

Here, (fig-3) represents the basic flow of the whole project, its interaction with the user to the end process.

Each of the operation can also be used with the shortcut key specified to each of them -

('Addition', '+', '**Control-Shift-A**')
('Subtraction', '-', '**Control-Shift-S**')
('Multiplication', 'x', '**Control-Shift-M**')
('Division', '/', '**Control-Shift-D**')
('Square root', '√', '**Control-Shift-Q**')
('Primitive root', '∝', '**Control-Shift-R**')
('XOR', 'XOR', '**Control-Shift-X**')
('Module inverse', 'Inv', '**Control-Shift-I**')
('Exponentiation', 'a^b', '**Control-Shift-E**')
('Module', 'mod', '**Control-Shift-N**')
('GCD', 'GCD', '**Control-Shift-G**')
('LCM', 'LCM', '**Control-Shift-L**')
('Primality', 'Prime', '**Control-Shift-P**')
('Factorization', 'Fact', '**Control-Shift-F**')
('Discrete Logarithm', 'DLP', '**Control-Shift-H**')

IV. RESULTS AND DISCUSSIONS

We have the final output as a gui, made by tkinter of a calculator, more precisely a modulo calculator having modulo operations on addition, subtraction, gcd, discrete logarithm, etc as mentioned earlier. We have the option to either include the module feature by simple checking out the checkbox or we even have the option to exclude it. At the right side, we have history box, showing all the calculations, we have performed earlier. This history is also saved in a text file known as log file.

V. FUTURE SCOPE

The Modular Arithmetic field itself has very large extensions and strong application in Cryptography. One of the main reasons is that modular arithmetic makes it simple to form groups, rings, and fields, which are crucial components of most modern public-key cryptosystems. This project is Modular Arithmetic Calculator. Here, for now just the Base-10 is used, i.e. for now it is only applicable for decimal numbers, but it can be extended to base-2, base-8 and base-16 as well, i.e. binary, octal and hexadecimal as well. This project can also be hosted as an application on phones which contains so many features of operations with mod, which is quite different and unique from the normal calculator.

VI. CONCLUSION

Modular arithmetic is an example of a new notion being defined through abstraction from an old one, in this case integer arithmetic. On the integers, we construct an "equivalence relation," and on the "equivalence classes," we define arithmetic. Although modular arithmetic has a variety of practical uses, offering examples for a mathematician is like giving examples for a fundamental tool like a lathe.

Modular arithmetic is a fundamental tool for us, especially in number theory. A lot of check-sums use modular arithmetic. For example, the digits in a book's ISBN (international standard book number) satisfy a congruence relation. It's more difficult to make a typo in the book number without the congruence being thrown off (letting one know that a mistake has been made). There is a maximum number in modular arithmetic. Clock arithmetic is the most well-known example. In the United States, we don't say 13 o'clock; instead, we start over and call it 1 O'clock. We add 7, divide by 12, and preserve the remaining if the time is 9 and we want to know what time it will be 7 hours later. (9+7=16, with a remainder of 4).

Similarly, we might only have the integers 0, 1, and 2 if we're conducting arithmetic modulo 3. 1+1 = 2, 2+1 = 0, 2*2 =1, 2*2 =1, 2*2 =1, and so on. Another application is in cryptocurrency or Internet security in general. It is built on the foundations of number theory and modular arithmetic. There is no Internet commerce or bitcoin without modular arithmetic.

REFERENCES

[1] Johann Großschädl - “Energy-efficient Software implementation of long integer modular arithmetic” 2005 [2] Xiaotian Wu - “Improving recovered image quality in secret image sharing by simple modular arithmetic” 2018 [3] M. K. Stojčev - “A unified approach in manipulation with modular arithmetic” may 2012

[4] R.Chaves 1 and l.sousa 1 - “improving residue number system multiplication with more balanced moduli sets and enhanced modular arithmetic structures”

[5] Thomas Plantard - “efficient word size modular arithmetic” sep 2021

[6] P. G. Comba -“ exponentiation cryptosystems on the ibm” dec. 1990.

[7] S. R. Duss´e and b. s. kaliski.- “a cryptographic library for the motorola in advances in cryptology”

[8] J. R. Goodman - “energy scalable reconfigurable cryptographic hardware for portable applications.” 2000 [9] Colin D. Walter- “data integrity in hardware for modular arithmetic” january 2002

[10]https://www.irishtimes.com/news/science/modulararithmetic-you-may-not-know-it-but-you-use-it-every-day-1.3268649

[11] https://en.wikipedia.org/wiki/modular_arithmetic\

[12] https://www.geeksforgeeks.org/modular-arithmetic/

[13] https://brilliant.org/wiki/modular-arithmetic/

[14] Vayadande, Kuldeep, Ritesh Pokarne, Mahalaxmi Phaldesai, Tanushri Bhuruk, Tanmai Patil, and Prachi Kumar. "SIMULATION OF CONWAY’S GAME OF LIFE USING CELLULAR AUTOMATA." International Research Journal of Engineering and Technology (IRJET) 9, no. 01 (2022): 2395-0056.

[15]Vayadande, Kuldeep, Ram Mandhana, Kaustubh Paralkar, Dhananjay Pawal, Siddhant Deshpande, and Vishal Sonkusale. "Pattern Matching in File System." International Journal of Computer Applications 975: 8887.

[16]Vayadande, Kuldeep, Neha Bhavar, Sayee Chauhan, Sushrut Kulkarni, Abhijit Thorat, and Yash Annapure. Spell Checker Model for String Comparison in Automata. No. 7375. EasyChair, 2022.

[17] VAYADANDE, KULDEEP. "Simulating Derivations of Context-Free Grammar." (2022).

[18]Vayadande, Kuldeep, Neha Bhavar, Sayee Chauhan, Sushrut Kulkarni, Abhijit Thorat, and Yash Annapure. Spell Checker Model for String Comparison in Automata. No. 7375. EasyChair, 2022

[19] Varad Ingale, Kuldeep Vayadande, Vivek Verma, Abhishek Yeole, Sahil Zawar, Zoya Jamadar. Lexical analyzer using DFA, International Journal of Advance Research, Ideas and Innovations in Technology, www.IJARIIT.com.

[20]Vayadande, Kuldeep, Harshwardhan More,Omkar More, Shubham Mulay,Atahrv Pathak, Vishwam Talanikar, “Pac Man: Game Development using PDA and OOP”, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056, p-ISSN: 2395-0072, Volume: 09 Issue: 01 | Jan 2022, www.irjet.net

[21]Vayadande, Kuldeep, Parth Sheth, Arvind Shelke, Vaishnavi Patil, Srushti Shevate, Chinmayee Sawakare, “Simulation and Testing of Deterministic Finite Automata Machine,” *International Journal of Computer Sciences and Engineering*, Vol.10, Issue.1, pp.13-17, 2022.

[22]Rohit Gurav, Sakshi Suryawanshi,Parth Narkhede,Sankalp Patil,Sejal Hukare,Kuldeep Vayadande,” Universal Turing machine simulator”, International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X, (Volume 8, Issue 1 - V8I1-1268, https://www.ijariit.com/

[23]Vayadande, Kuldeep, Krisha Patel, Nikita Punde, Shreyash Patil, Srushti Nikam, Sudhanshu Pathrabe, “Non-Deterministic Finite Automata to Deterministic Finite Automata Conversion by Subset Construction Method using Python,” *International Journal of Computer Sciences and Engineering*, Vol.10, Issue.1, pp.1-5, 2022.

[24]Vayadande, K. B., and Nikhil D. Karande. "Automatic Detection and Correction of Software Faults: A Review Paper." International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653.