Spl-1 Project Report- 2023

**Numeric Operations Library**

Supervised By

Abdus Satter  
Assistant Professor,  
Institue Of Information Technology

Submitted by:

Salsabila Zaman  
BSSE Roll:1443

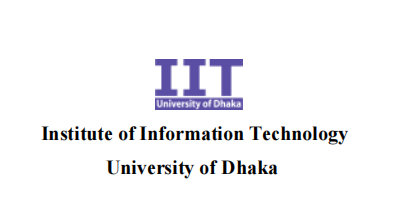


Table of Contents

4 Introduction

5 Domain and Overview

6 Background

7 Description

-Implementation

-Testing(Sample input output)

-Code Snippet

8 Challenges Faced Describe the challenges you faced and how have you overcome those.

9 Conclusion What have you learnt? How can the project be extended in future?

10 References [1] URL, Name of the page, last accessed on dd MMM YYYY

[2] Academic Paper title, author names, conference / journal it was published, year, page

List Of Figures

1. Introduction

Write the broad domain and overview of your project.

2. Background of the Project

Describe the topics that are related to your project. For example, you may present the

definition and example that a reader must know to understand your project. You must add

appropriate references.

3. Description of the Project

Describe your work. Add flowcharts if necessary.

4. Implementation and Testing

Insert implementation details and testing details (if applicable). You may add tables/diagrams/

code snippet if needed.

5. User Interface

Sample input and output. You may add screenshots if required.

6. Challenges Faced

Describe the challenges you faced and how have you overcome those.

7. Conclusion

What have you learnt? How can the project be extended in future?

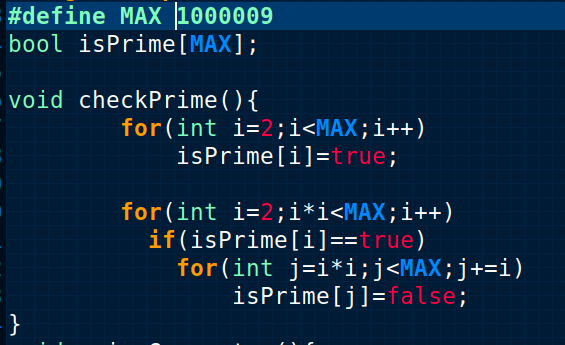
**Prime Numbers**

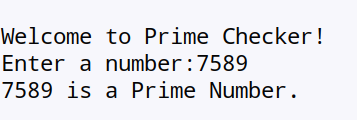
A prime number is a whole number greater than 1 whose only factors are 1 and itself. Here in my project, you can check

-Check if a number is prime or not

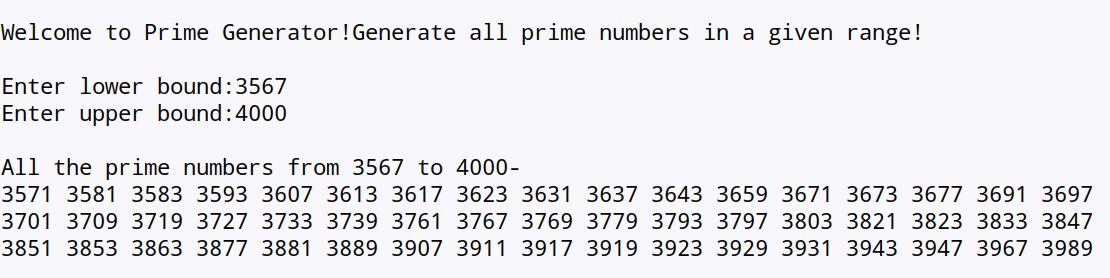
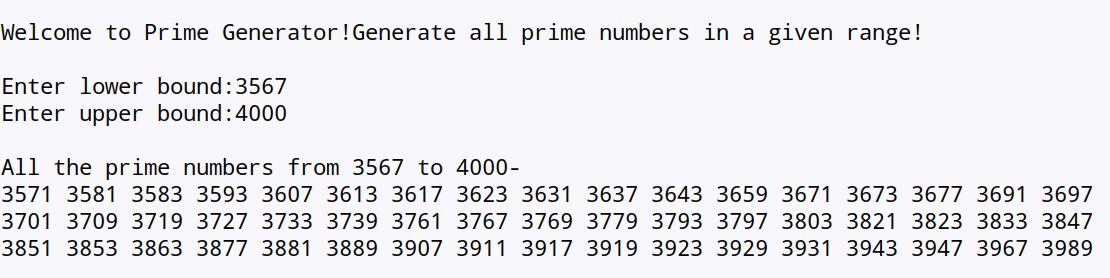
- Generate all prime numbers in a given range

For both of the work, I implemented Sieve of Erathosthnenes and precalculated all prime numbers upto 1000009.Sieve of Eratosthenes is an almost mechanical procedure for separating out composite numbers and leaving the primes.



Figure 1: Prime Checking

At the beginning this code first marks all numbers except zero and one as potential prime numbers, then it begins the process of sifting composite numbers. For this it iterates over all numbers from 2  to n . If the current number i  is a prime number, it marks all numbers that are multiples of i  as composite numbers, starting from i^2 . This is already an optimization over naive way of implementing it, and is allowed as all smaller numbers that are multiples of i  necessary also have a prime factor which is less than i , so all of them were already sifted earlier.



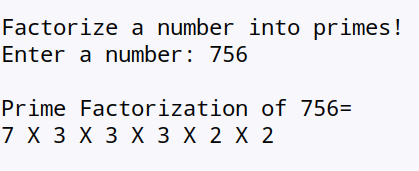
Although the sieve can be a tedious process for discovering large primes,it is a procedure that can be effectively turned over to a computer, using a language such as Fortran, BASIC, or Pascal. According to Ore, every table of primes has been constructed with the method described by Eratosthenes. This includes tables of all the primes up to one hundred million.

**Prime Factorization**

Prime factorization is a process of factoring a number in terms of prime numbers i.e. the factors will be prime numbers. Here we used the Division Method to calculate the prime factors. The steps to calculate the prime factors of a number is similar to the process of finding the factors of a large number.

Step 1: Divide the given number by the smallest prime number. In this case, the smallest prime number should divide the number exactly.

* Step 2: Again, divide the quotient by the smallest prime number.
* Step 3: Repeat the process, until the quotient becomes 1.
* Step 4: Finally, multiply all the prime factors



Digit Manipulations

Digit manipulation involves operations or algorithms that target specific digits within a numerical representation, such as extracting, modifying, or analyzing individual digits based on their position. Here the goal is to perform operations on individual digits of integer values.

Digit Analysis Tool:

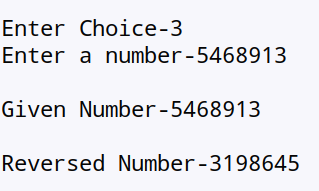
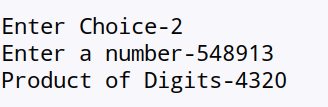
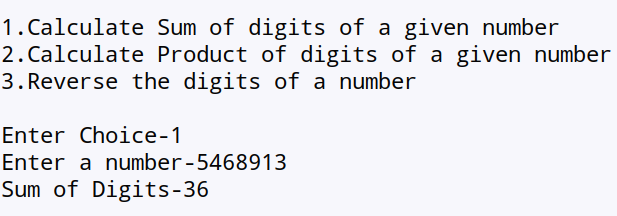
* The code can be part of a larger program that analyzes and manipulates digits in various ways.
* It could be used in applications where digit operations are necessary, such as in mathematical calculations, data validation, or encryption algorithms.
* User Input Processing:
* The digit manipulation functions can be utilized in scenarios where user input involves numerical data.
* For example, in a program that processes and validates user-entered numbers, the code could be used to extract information about the digits.

Password Security:

* In password creation and authentication systems, digit manipulation may be used to ensure password strength. For instance, algorithms that check for the presence of numbers, their position, or their frequency can enhance password security.

Data Analysis:

* In data analysis, digit manipulation is crucial for processing and extracting relevant information from numerical datasets. For example, analyzing sales figures, population statistics, or any numerical data may involve digit manipulation.

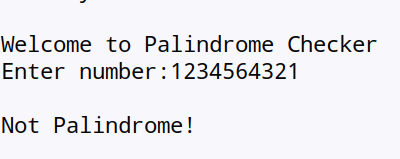
Palindrome Checker

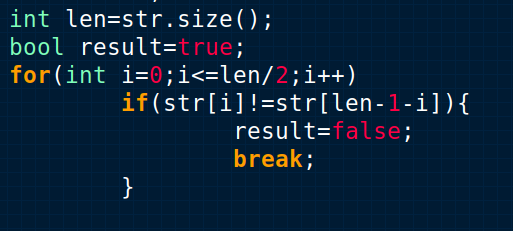
A **palindrome** is a word, [number](https://en.wikipedia.org/wiki/Palindromic_number), phrase, or other sequence of symbols that reads the same backwards as forwards, such as *madam* or *racecar*,A palindrome checker is a program or algorithm that determines whether a given sequence of characters forms a palindrome.

Developing an efficient palindrome checker can be a starting point for exploring different algorithms and optimization techniques in computer science.

1. Security and Cryptography:
   * Palindromes may be used in certain cryptographic applications or security algorithms, and a palindrome checker could be employed to ensure the integrity of a sequence.
2. Fun and Recreational Programming:
   * Creating a palindrome checker can be a fun programming exercise. It's a simple problem that allows programmers to practice their skills and experiment with different approaches.
3. Data Filtering:
   * In scenarios where alphanumeric data needs to be filtered or processed, a palindrome checker can help identify palindromic patterns within the data.

While the palindrome checker itself is a simple concept, its application can be diverse. It provides a foundation for understanding and implementing basic string manipulation and comparison algorithms.





GCD Calculator

The greatest common divisor (GCD) of two numbers is the greatest factor that divides both the numbers. Here I used Euclidean algorithm.

Formal description of the Euclidean algorithm

* **Input** Two positive integers, a and b.
* **Output** The greatest common divisor, g, of a and b.
* **Internal computation**
  1. If a<b, exchange a and b.
  2. Divide a by b and get the remainder, r. If r=0, report b as the GCD of a and b.
  3. Replace a by b and replace b by r. Return to the previous step.

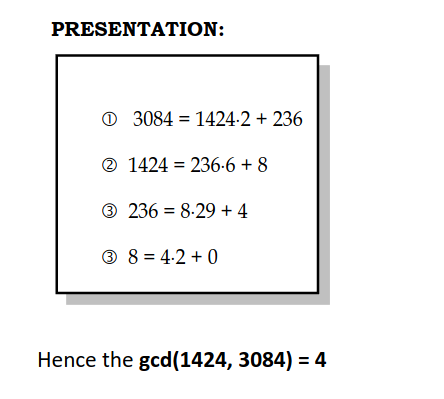
Divide 210 by 45, and get the result 4 with remainder 30, so 210=4**·**45+30.

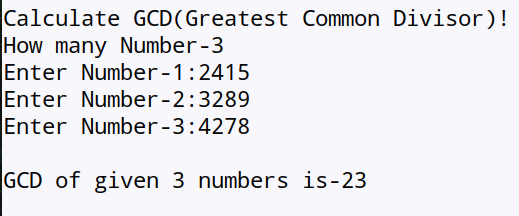
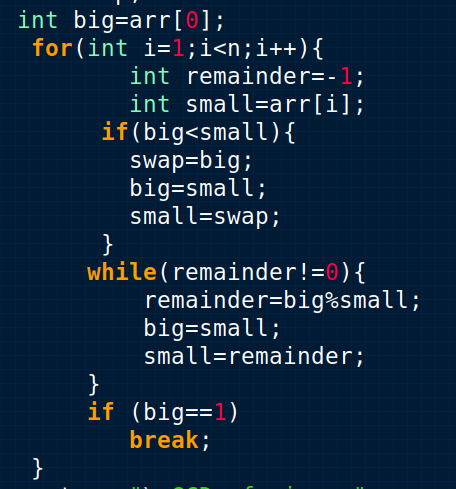
Divide 45 by 30, and get the result 1 with remainder 15, so 45=1**·**30+15.

Divide 30 by 15, and get the result 2 with remainder 0, so 30=2**·**15+0.

The greatest common divisor of 210 and 45 is 15.

Here I extended the euclidean algorithm for multiple numbers. Firstly,we calculated gcd for two numbers and then the ran the same algorithm on gcd of previous two numbers and another number ,until no numbers are left.

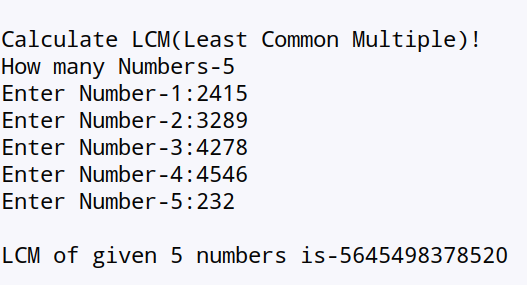
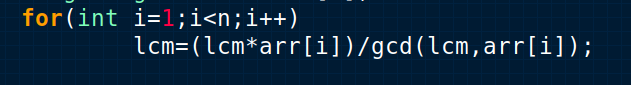


LCM Calculator

LCM ([Least Common Multiple](https://www.geeksforgeeks.org/least-common-multiple/)) of two numbers is the smallest number which can be divided by both numbers.Here we calculated LCM using GCD using this formula.

LCM(a, b) = (a x b) / GCD(a, b)

Fibonacci Sequences

The Fibonacci sequence is a set of integers (the Fibonacci numbers) that starts with a [zero](https://www.techtarget.com/whatis/definition/zero-0), followed by a one, then by another one, and then by a series of steadily increasing numbers. The sequence follows the rule that each number is equal to the sum of the preceding two numbers.

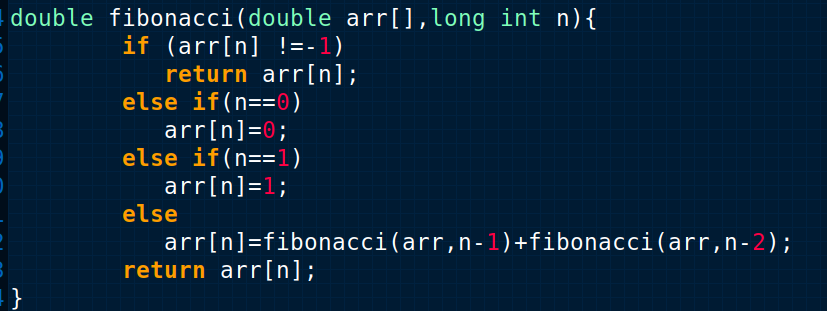
The Fibonacci sequence begins with the following 14 integers:

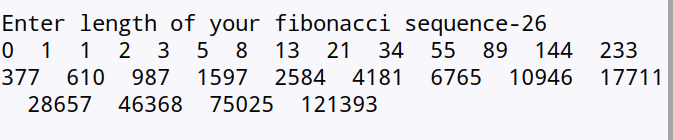
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233 …

the Fibonacci sequence can be defined by the following three equations:

* F0 = 0 (applies only to the first integer)
* F1 = 1 (applies only to the second integer)
* Fn = Fn-1 + Fn-2 (applies to all other integers)

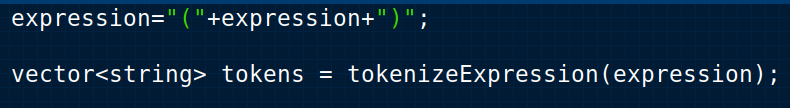
Here I generated fibonacci sequence of numbers with dynamic programming



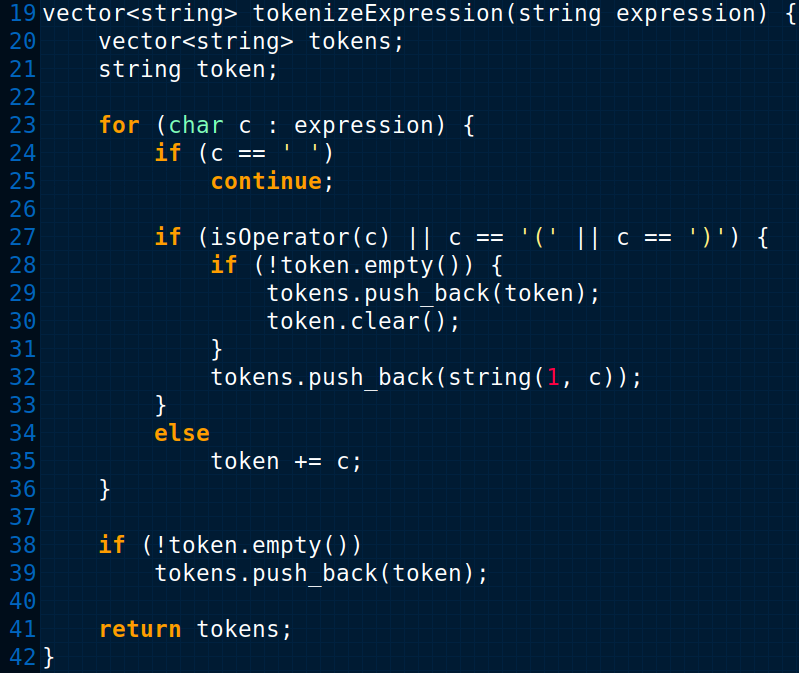
Expression Evaluation

What is an expression?

An expression is a combination of operators, constants and variables. An expression may consist of one or more operands, and zero or more operators to produce a value. There are various types of expressions- Arithmetic,Logical,Relational,Conditional. We are here working with arithmetic expressions. An *arithmetic expression* is an expression using additions **+**, subtractions **-**, multiplications **\***, divisions **/**, and exponentials **^**.A parenthesis can be used to override the precedence of operators and force the evaluation of a sub-expression within an expression.

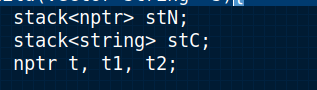


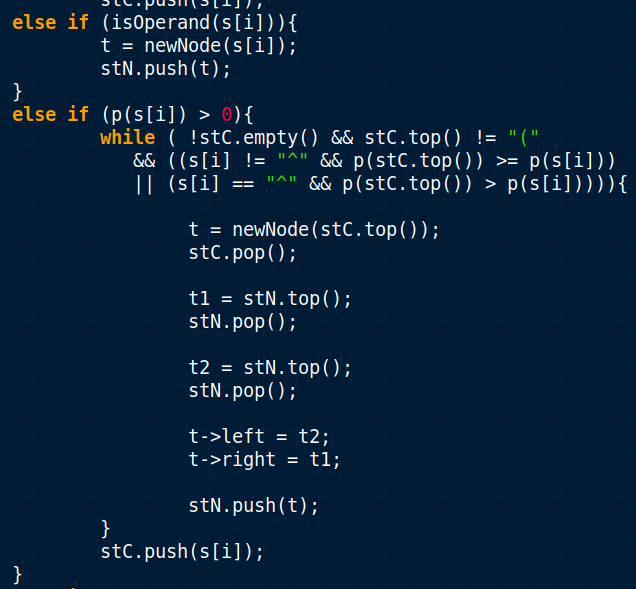
The program is taking input the expression from the user and encapsulating it with brackets,the left and right bracket will work as a marker for the start and end of the expression(string). Then the program tokenizes the expression and convert it into a vector of strings,where each string is a token(operator/operand).

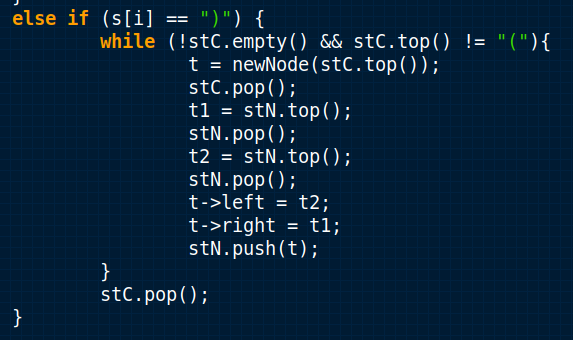


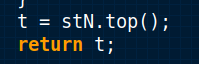
We read the expression character by character using for-loop,it ignores whitespace. It also keeps a string named “token”,which is to put all parts of an operand to a single operand. Example- if there is an operand 3.1416, firstly the token=3,then token=3. ,then token=3.14 ,it keeps adding until an operator is found. When an operator is found, it pushes the current token into tokens vector and empties the current token and then the operator is also push into the vector. Last if is for the last operand to be push on to the vector. Lastly it returns the vector containing tokens.

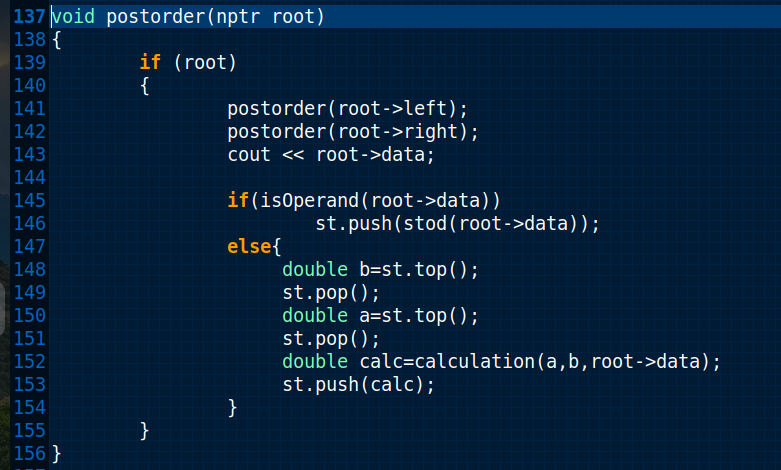
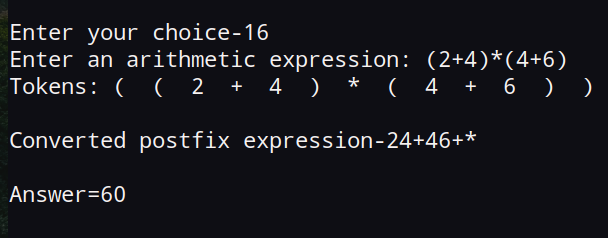
Implemented the Shunting Yard Algorithm to build a binary expression tree from an infix expression represented as a vector of strings  
The idea behind this condition is to pop operators from the stack and process them when the operator at the top of the stack has higher precedence than or equal to the current operator (s[i]). This ensures that operators with higher precedence are processed first, following the rules of the Shunting Yard Algorithm.







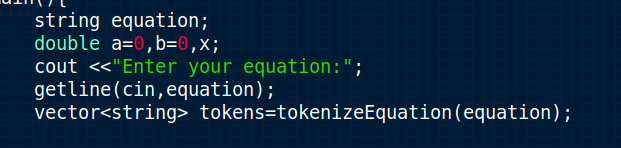


**Linear Equation Solver(One Variable)**

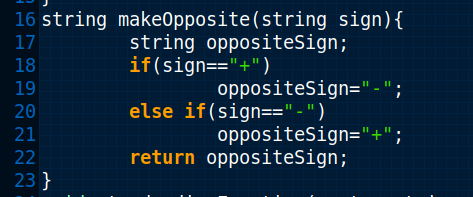
An equation that has the highest degree of 1 is known as a linear equation.A linear equation in one variable is an equation in which there is only one variable present. It is of the form

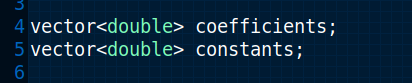
**Ax+B=0,**where A and B are any two real numbers and x is an unknown variable that has only one solution.

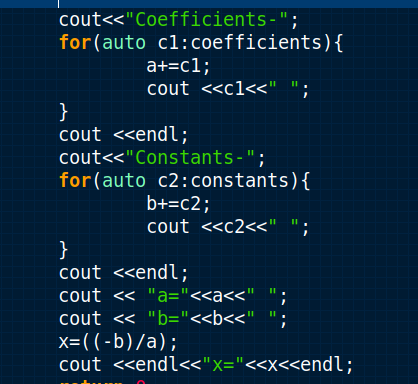
****

****

****

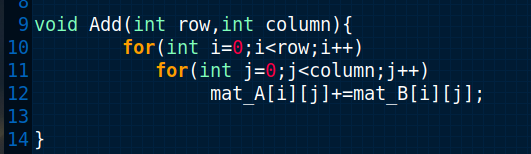
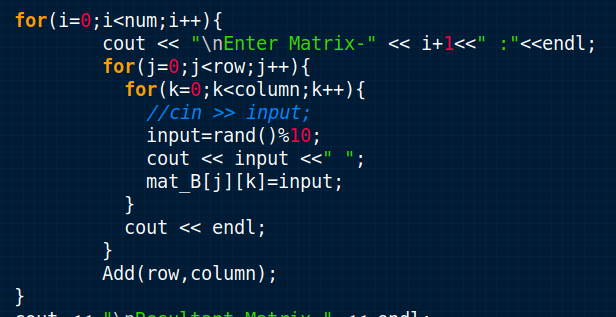
****

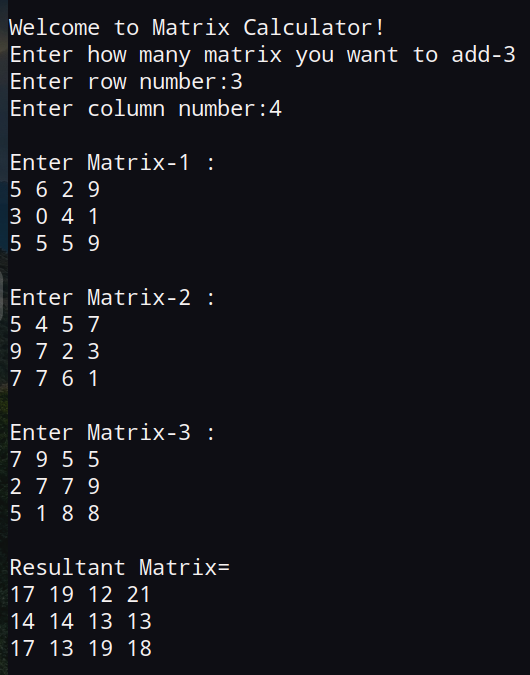
****

****

**Matrix Addition**

The addition of matrices is an operation on matrices where corresponding elements of two or more matrices are added. Matrices can be added only if they are of the same size, that is, they have the same dimension or order.





Matrix Multiplication

The product of two matrices A and B is defined if the number of columns of A is equal to the number of rows of B. Let A = [aij] be an m × n matrix and B = [bjk] be an n × p matrix. Then the product of the matrices A and B is the matrix C of order m × p. To get the (i, k)th element c of the matrix C, we take the ith row of A and kth column of B, multiply them element-wise and take the sum of all these products.

*A**B*=[*c**i**j*] , where *c**i**j*=*a**i*1*b*1*j*+*a**i*2*b*2*j*+...+*a**i**n**b**n**j*

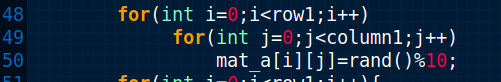
(The entry in the *i*th

row and *j*th column is denoted by the double subscript notation *a**i**j* , *b**i**j* , and *c**i**j* . For instance, the entry *a*23

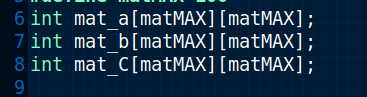
is the entry in the second row and third column.)

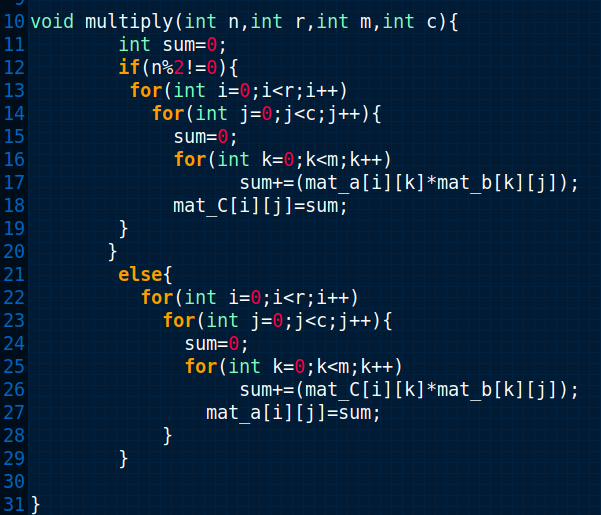
The definition of matrix multiplication indicates a row-by-column multiplication, where the entries in the *i*th

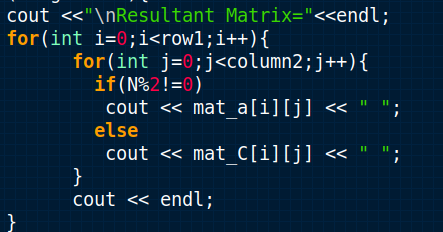
row of *A* are multiplied by the corresponding entries in the *j*th column of *B* and then adding the results.

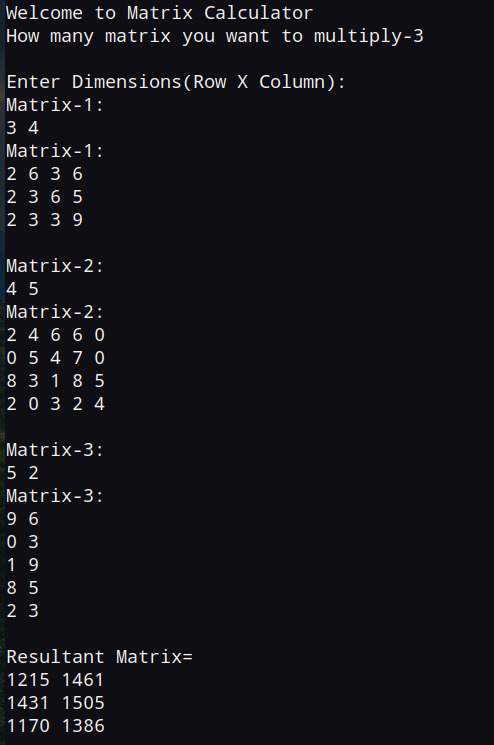










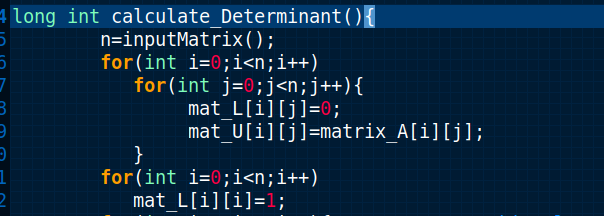


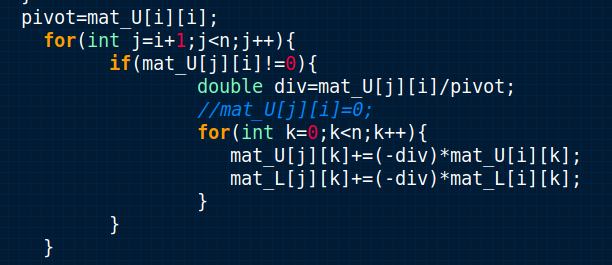
Matrix Determinant

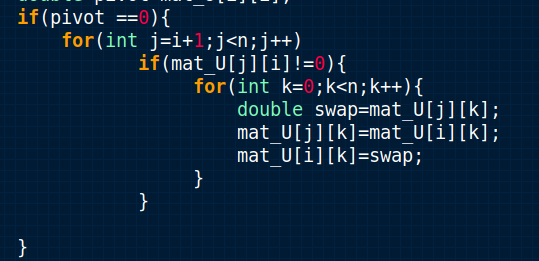
The determinant of a matrix is a single numerical value which is used when calculating [the inverse](https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/core-mathematics/pure-maths/matrices/matrix-inverse.html) or when solving systems of linear equations.

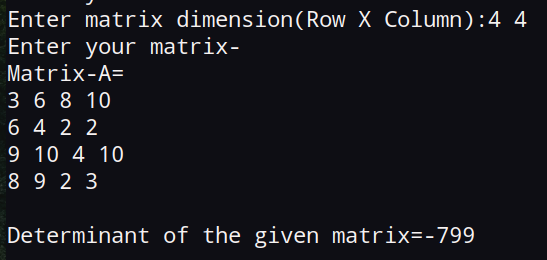
The determinant of a matrix A

is denoted |A|, or sometimes det(A). The determinant is only defined for square matrices.





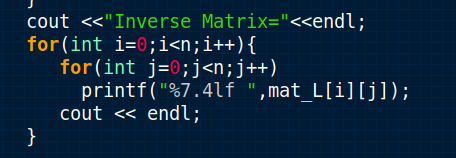
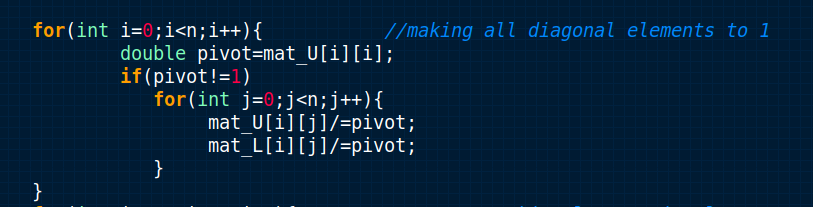


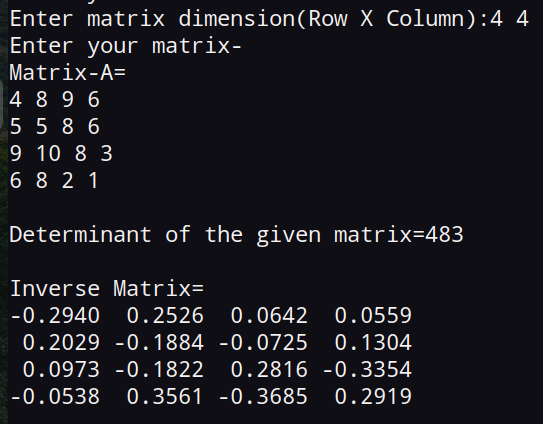


Matrix Inversion

Inverse of a Matrix is a matrix that on multiplying with the original matrix result in the identity matrix. It is required to solve complex problems using matrix operations. For any matrix A its inverse is denoted as A-1. Inverses only exist for square matrices.

Gauss-Jordan Elimination method





Matrix Power Calculation

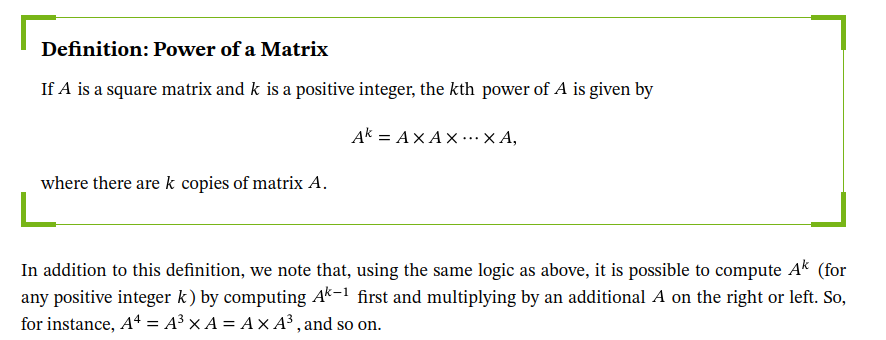
Matrix power is obtained by multiplication matrix by itself 'n' times.

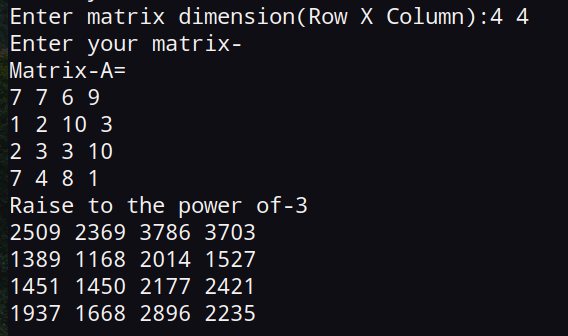
1. The matrix must be square in order to raise it to a power.

### Definition: Power of a Matrix

If is a square matrix and is a positive integer, the power of is given by where there are copies of matrix .

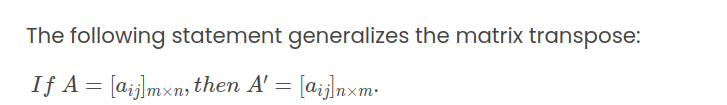
In addition to this definition, we note that, using the same logic as above, it is possible to compute (for any positive integer ) by computing first and multiplying by an additional on the right or left. So, for instance, , and so on.

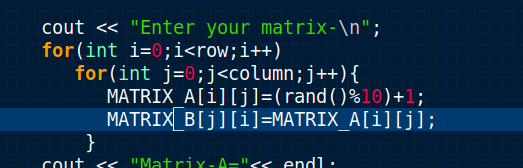




Matrix Transposition

Transpose of a Matrix is defined as “A Matrix which is formed by turning all the rows of a given matrix into columns and columns into rows..”





Matrix Eigenvalues and Eigenvectors

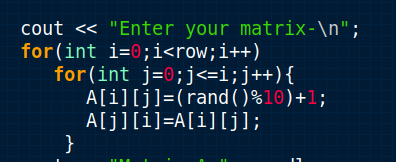
Eigenvalues are the special set of scalars associated with the system of linear equations. It is mostly used in matrix equations.The basic equation is

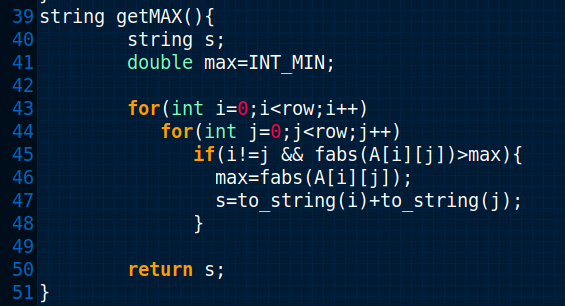
**Ax = λx**

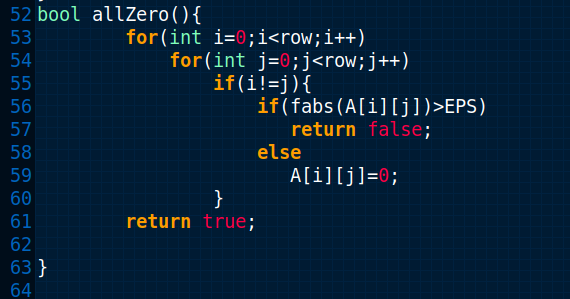
The number or scalar value “λ” is an eigenvalue of A.

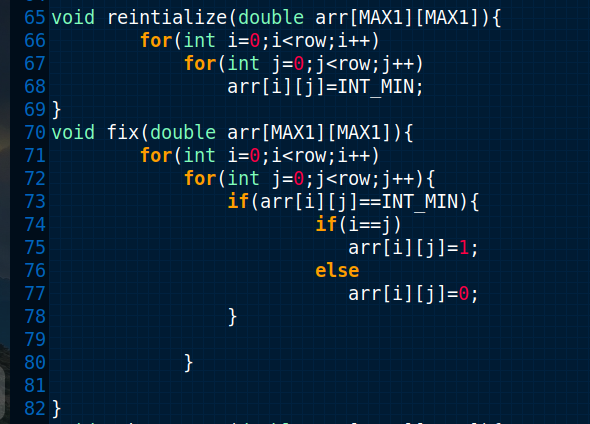
In Mathematics, an eigenvector corresponds to the real non zero eigenvalues which point in the direction stretched by the transformation whereas eigenvalue is considered as a factor by which it is stretched. In case, if the eigenvalue is negative, the direction of the transformation is negative.

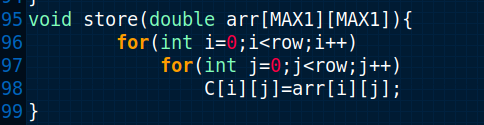
For every real matrix,  there is an eigenvalue. Sometimes it might be complex

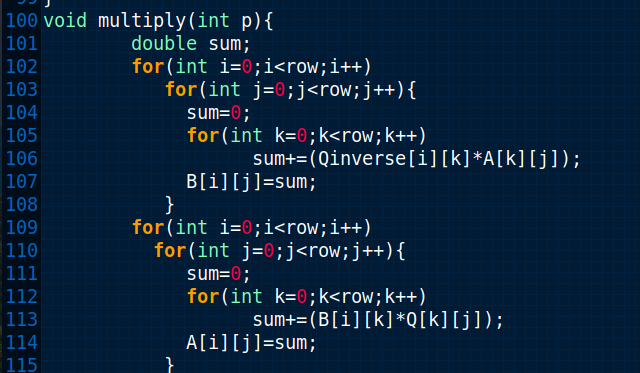


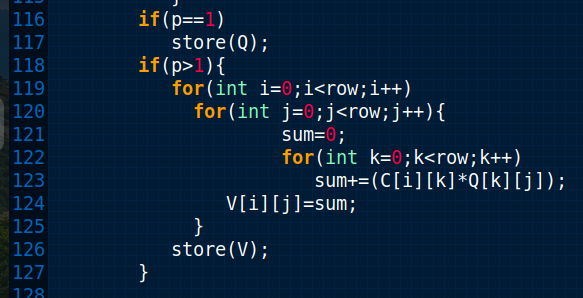


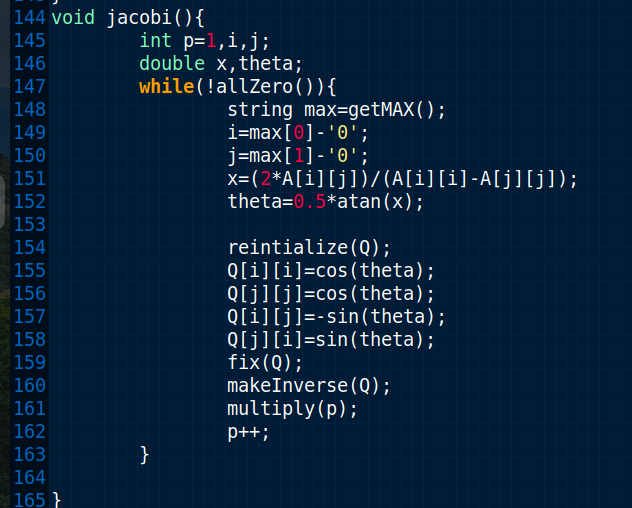












Matrix Rank

The rank of the matrix refers to the number of linearly independent rows or columns in the matrix.

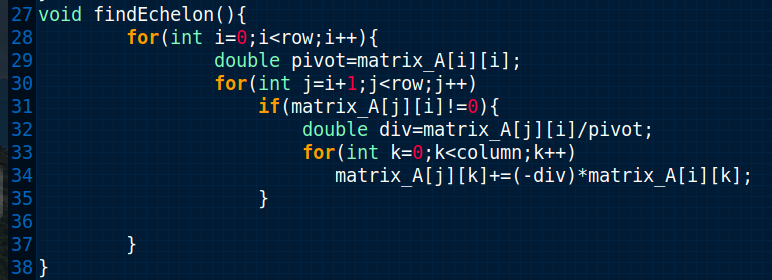
Find the Rank of a Matrix Using the Echelon Form.A matrix 'A' is said to be in Echelon form if it is either in upper triangular form or in lower triangular form. We can use elementary row/column transformations and convert the matrix into Echelon form.

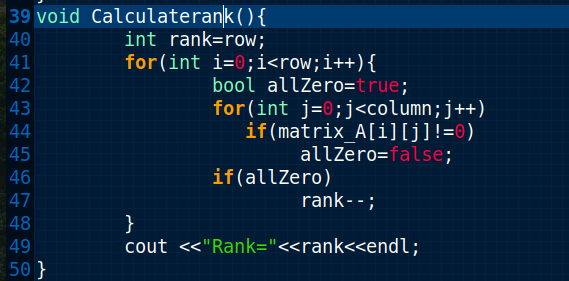
A row (or column) transformation can be one of the following:

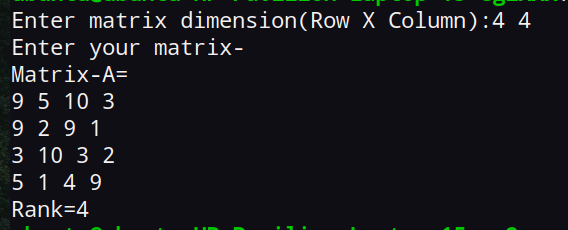
* Interchanging two rows.
* Multiplying a row by a scalar.
* Multiplying a row by a scalar and then adding it to the other row.

Here are the steps to find the rank of a matrix.

* Convert the [matrix](https://www.cuemath.com/algebra/solve-matrices/) into Echelon form using row/column transformations.
* Then the rank of the matrix is equal to the number of non-zero rows in the resultant matrix.







References

https://www.encyclopedia.com/science/encyclopedias-almanacs-transcripts-and-maps/sieve-eratosthenes-0

https://byjus.com/maths/prime-factorization/

https://www.khanacademy.org/computing/computer-science/cryptography/modarithmetic/a/the-euclidean-algorithm

[https://www.idomaths.com/hcflcm.php#google\_vignette](https://www.idomaths.com/hcflcm.php" \l "google_vignette)

https://www.prepbytes.com/blog/stacks/expression-evaluation/

https://brilliant.org/wiki/shunting-yard-algorithm/

https://www.cuemath.com/algebra/addition-of-matrices/

https://www.toppr.com/guides/maths/matrices/multiplication-of-matrices/

https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/core-mathematics/pure-maths/matrices/matrix-determinant.html

https://people.richland.edu/james/lecture/m116/matrices/inverses.html

https://www.nagwa.com/en/explainers/432180315293/

https://www.vedantu.com/maths/matrix-rank

https://www.cmi.ac.in/~ksutar/NLA2013/iterativemethods.pdf