

# MATRIX CALCULATOR

## Requirements :-

### Introduction :-

Mathematical operations are a part of our daily lives. Everyday we will be involving with various types of calculations around us. Matrices is a mathematical entity where numbers are arranged in rows and columns. A lot of applications exist with matrices in real life. The major application lies in the software industry such as development of algorithms like path finder algorithms, image processing algorithms and many more. In this project, some of the basic matrix operations are presented where a user can select the operation to be performed on the matrix. Then the matrices with their size are entered. Note that only square matrices are being considered for the project.

### Research :-

**Objective** :- This project focuses on matrix operations on square matrices. The operations performed on a pair of square matrices are, addition, subtraction and multiplication. The operations performed on a single square matrix are, transpose, determinant and inverse.

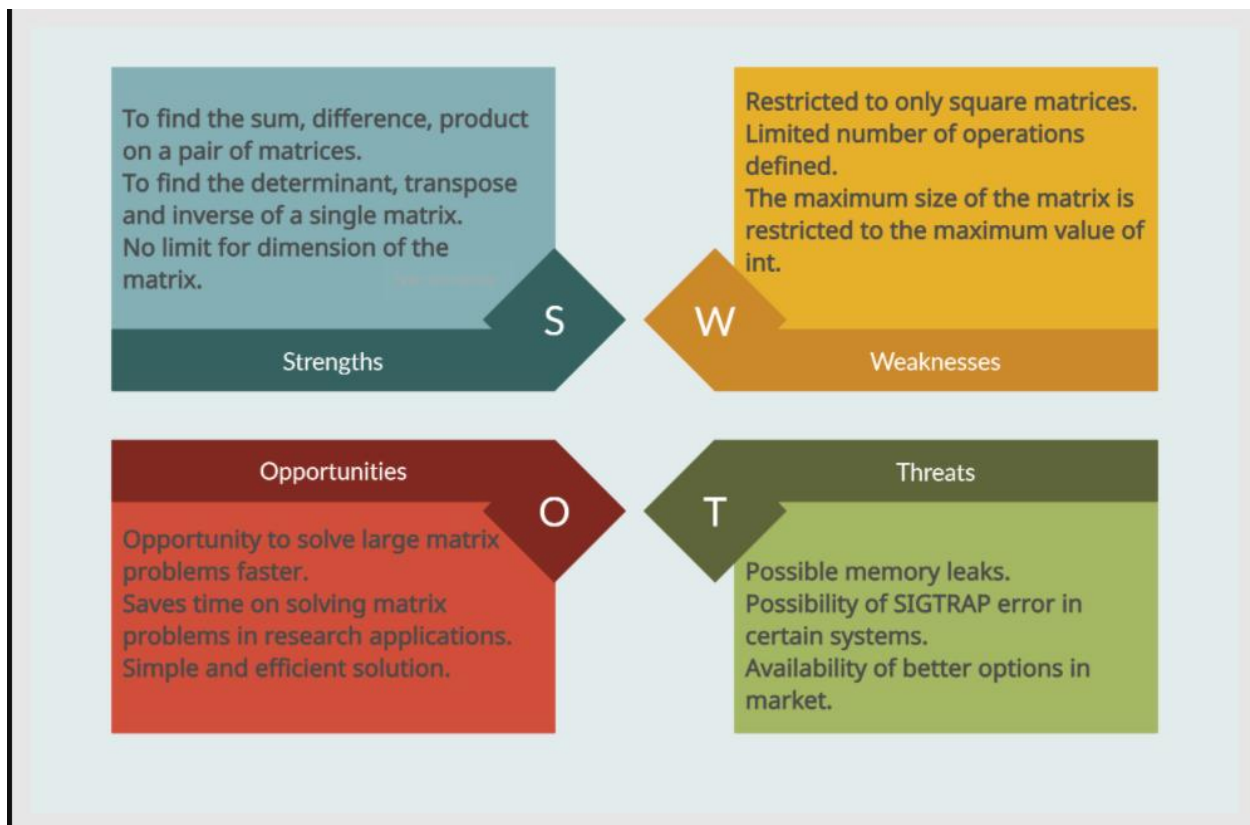
**Benefits**:- A user who is working on a laptop or a desktop who would be working on matrix based calculations, can make use of this application as handy. A human being can find the output values of a matrix for a maximum of size  $3 \times 3$ , i.e 3 rows and 3 columns. If they try to solve the matrix for dimensions above 3, the time consumed is large and verification that the solution is correct or not consumes extra time. Hence this tool can be used to cross check the matrix operations conducted.

**Cost and Features**:- The cost for implementing a project involving matrix calculations is minimal to none. But the extension of matrix applications to various state of the art domains like machine learning, image processing, cryptography, path finder algorithms etc will have a research and development being done. And for the required calculations related to matrices, this program comes in handy.

**Some of The features of the project**:- This program has dynamic memory allocation involved for allocating memory for the matrices, i.e 2 dimensional arrays. A structure is implemented for storing the input matrices. The user can select the operation to be performed by pressing a number on the keyboard. The significance of each number, i.e the operation which it performs will be displayed on the screen during the execution of the program.

**Defining Our System:-** The system on the whole is viewed as a blackbox which accepts input of a single number. Then the mapping of the number is done so as to perform the appropriate function. Then the user will be prompted to enter the matrix size. After that the user is prompted to enter the matrix elements. The input is stored in 2D arrays which are dynamically allocated and passed to the respective functions. Then the output is printed on the screen. This whole process is in a loop until the user presses a key which will exit the program.

## Swot analysis:-



## 4W's and 1'H

### Who:

Students who want to solve matrix related problems for their academic work  
Datascience workers working on machine learning and deep learning applications which majorly involve matrices. Software engineers Researchers and mathematicians

### What:

Calculate the sum, difference, product of two matrices and also the determinant, transpose and inverse of a single matrix.

**When:**

Students facing a difficulty in solving a matrix related problem, can use this program to confirm their outputs. Researchers can use it for getting a quick calculation output. Comes to a great use when calculating for matrices of huge order.

**Where:**

Students, employees and researchers all over the world.

**How:**

This program can be executed in a system which has Linux or Windows operating system.

**High Level Requirements:**

ID	Description	Category	Status
HR01	User shall be able to specify the size of the matrix	Technical	IMPLEMENTED
HR02	User shall be able to add the matrices	Technical	IMPLEMENTED
HR03	User shall be able to subtract the matrices	Technical	IMPLEMENTED
HR04	User shall be able to multiply the matrices	Technical	IMPLEMENTED
HR05	User shall be able to find determinant, transpose and inverse of a matrix	Technical	IMPLEMENTED

## Low level Requirements:

ID	Description	HLR ID	Status (Implemented/Future)
LR01	(1). User must specify the size of the matrix. (2). The size specified is the dimensions of the square matrix	HR01	IMPLEMENTED
LR02	Addition of two matrices can be done (1). Both the matrices need to be of same size(2). The size entered in the beginning is same for both the matrices	HR02	IMPLEMENTED
LR03	Subtraction of two matrices is implemented in the same manner as addition	HR03	IMPLEMENTED
LR04	Multiplication of two matrices is done on two square matrices of a specified size	HR04	IMPLEMENTED
LR05	The determinant of a single matrix of a specified size is implemented	HR05	IMPLEMENTED
LR06	The transpose of a single matrix operation is implemented	HR05	IMPLEMENTED
LR07	To find the inverse of a matrix, both the determinant function and the transpose functions are used in it. The condition that a matrix is invertible is to check if the determinant of the matrix is equal to 0.	HR05	IMPLEMENTED

## FOLDER STRUCTURE:-

Folder	description
cov	.gcno and .c.gcov files for testing c program
doc	Doxygen configuration file which can generate documentation in html
inc	All header files
src	Main source code for calculator
test	All source code and data for testing purposes
build	Build output (Not included in git)

## FEATURES:-

Features	Issues Raised	Issues Resolved	No Test Cases	Test Case Pass
F_01, F_02, F_03, F_04, F_05, F_06, F_07	None	None	7	7

## FEATURES ID:-

Feature Id	Feature
F_01	Options to select matrix operation
F_02	Operations on two matrices such as addition, subtraction and multiplication are included
F_03	Single matrix operations such as determinant, transpose and inverse of a matrix
F_04	Separate function for each operation
F_05	A structure has been implemented for storing the matrices
F_06	Dynamic memory allocation and deallocation has been implemented for the matrices
F_07	There is no upper limit for the size of the matrix

### Challenges faced and how it was overcome:-

No.	Challenge	Solution
1.	Dynamic memory allocation of 2D arrays created segmentation faults	running the code in GDB helped find the line where the program crashes
2.	Program crashes	Writing clean code with allocating and deallocating memory at all functions as per requirement