

# Title: Matrix Calculator

**FIRST REVIEW DATE**  
**:24/12/2024**

Project Students:

NAME	REGISTER NO	BRANCH	SECTION
Sarvesh S	240701479	CSE	H
Sam Devaraja J	240701463	CSE	H

# Abstract

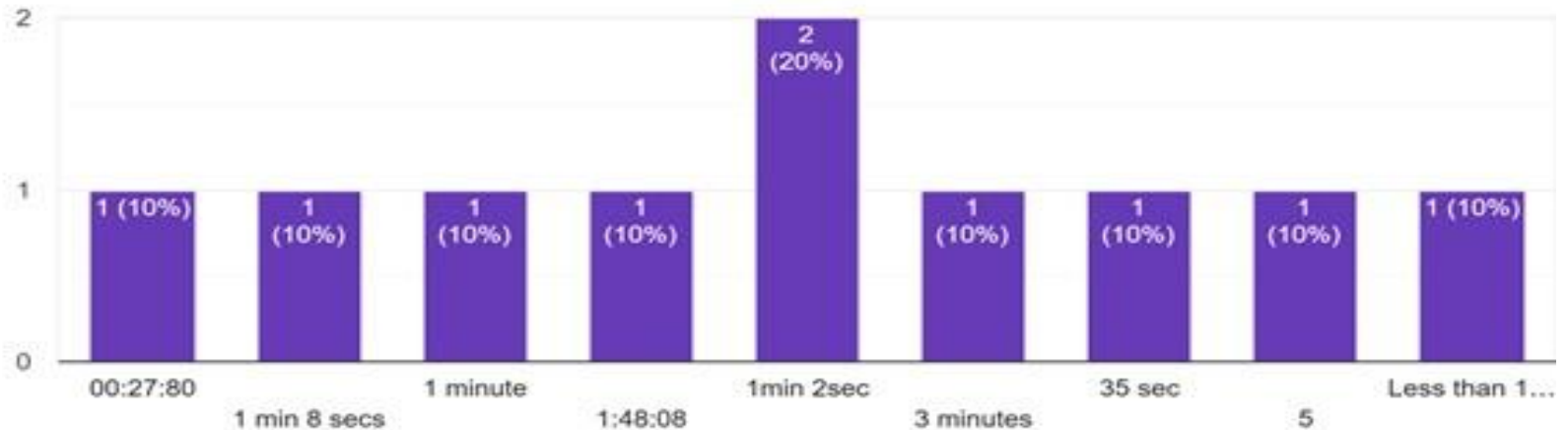
- This project implements a comprehensive Matrix Calculator using the C programming language, designed to perform various operations on 2x2 and 3x3 matrices. The calculator provides functionalities such as determinant calculation, matrix addition, subtraction, multiplication, transposition, and power computation. The program is structured for user interactivity, enabling the selection of specific operations through a menu-driven interface.
- Key features include:
  - Determinant Calculation: Computes the determinant of 2x2 and 3x3 matrices to determine linear independence.
  - Matrix Addition and Subtraction: Handles element-wise operations for 2x2 or 3x3 matrices.
  - Matrix Multiplication: Supports multiplication of matrices with compatible dimensions.
  - Transpose: Calculates the transpose of a 2x2 or 3x3 matrix.
  - Power of a Matrix: Raises square matrices to a user-specified power using repeated multiplication.
- The project showcases modular programming with distinct functions for each operation, emphasizing reusability and clarity. Error handling ensures input validity, guiding users in choosing appropriate matrix dimensions and operations. This calculator serves as a practical example of applying fundamental matrix operations, providing utility for educational purposes and basic computational needs.

# Literature Survey (If any)

Timestamp	Name	your determinant calculation Timing without calculator(minutes)	Your determinant calculation Timing with calculator(minutes)
2024/12/05 7:24:52 pm GMT+5:30	SANJAYRAM M	1:48:08	00:35:73
2024/12/05 7:26:50 pm GMT+5:30	P.Sahana	3 minutes	2 minutes
2024/12/05 7:30:40 pm GMT+5:30	SHERINKATHERINA.D	Less than 1 minute(45 second)	Less than 1 minute(25 seconds)
2024/12/05 7:31:02 pm GMT+5:30	Sahana k	1 minute	Less than one minute (30 seconds)
2024/12/05 7:41:32 pm GMT+5:30	Kirithick Raja	00:27:80	00:24:55
2024/12/05 7:49:09 pm GMT+5:30	SHARUKESH.D	1min 2sec	30 sec
2024/12/05 8:09:41 pm GMT+5:30	Roshni R	35 sec	25 sec
2024/12/05 8:25:37 pm GMT+5:30	Sankara Gomathi R	1 min 8 secs	44 secs
2024/12/05 8:45:26 pm GMT+5:30	Creator	1min 2sec	21 sec
2024/12/05 10:25:11 pm GMT+5:30	Rupashri	5	3

## your determinant calculation Timing without calculator(minutes)

10 responses



# Deviation from Existing Application

- **Customization for Educational Purpose:**
  - While many existing applications like MATLAB or online matrix calculators offer a wide range of advanced functionalities, this project focuses on fundamental matrix operations, specifically tailored for educational use and beginner-level programming practice.
- **Lightweight Implementation:**
  - Unlike sophisticated software requiring extensive resources, this program is lightweight and can run on any system with basic C compiler support, making it accessible for users with minimal computational power.
- **Interactive Console-Based Design:**
  - Unlike GUI-based applications, this project employs a command-line interface, providing a straightforward way to learn and understand the core mathematical computations and underlying logic.
- **No Dependency on External Libraries:**
  - Unlike many applications that rely on pre-built libraries, this project implements all operations from scratch, reinforcing the understanding of algorithms and matrix mathematics.

# Modules

- **int det2x2()** - Computes the determinant of a 2x2 matrix using the formula  $\det = a*d - b*c$ .
- Determines whether the matrix is linearly dependent or independent.
- **int det3x3()** - Calculates the determinant of a 3x3 matrix using the cofactor expansion method.
- Helps identify the matrix's properties, such as invertibility and dependency.
- **void matrixAddition()** - Performs element-wise addition of two matrices (2x2 or 3x3).
- Outputs the resultant matrix after addition.
- **void matrixSubtraction()** - Performs element-wise subtraction of one matrix from another (2x2 or 3x3).
- Outputs the resultant matrix after subtraction.
- **void matrixMultiplication()** - Multiplies two matrices of compatible dimensions and computes the resultant matrix.
- Supports arbitrary matrix sizes as long as the column count of the first matrix matches the row count of the second.
- **void transpose()** - Computes the transpose of a matrix (2x2 or 3x3) by swapping its rows and columns.
- Outputs the transposed matrix.
- **void power()** - Raises a square matrix (2x2 or 3x3) to a user-specified power by repeated multiplication.
- Outputs the matrix result after exponentiation.

# Deadlines

Sl.No.	Module Name	Target Date
1	det2x2()	22.11.2024
2	det3x3()	22.11.2024
3	matrixAddition()	01.12.2024
4	matrixSubraction()	01.12.2024
5	matrixMultiplication()	04.12.2024
6	transpose()	04.12.2024
7	power()	04.12.2024
	Product Completion	13.01.2025

# Expected Output

```
PS C:\Users\j_siv\OneDrive\Desktop\C\SEM-1(Mini project)> cd "c:\Users\j_siv\OneDrive\Desktop\C\SEM-1(Mini project)\\" ; if ($?) { gcc mainf.c -o mainf } ; if ($?) { .\mainf }  
-----"Welcome to matrix Calculator"-----
```

Select choice to calculate:

- 1) Determinant of 2x2 matrix
- 2) Determinant of 3x3 matrix
- 3) Matrix Addition (2x2 or 3x3)
- 4) Matrix Subtraction (2x2 or 3x3)
- 5) Matrix Multiplication (2x2 or 3x3)
- 6) Transpose of a matrix (2x2 or 3x3)
- 7) Power of a matrix (2x2 or 3x3)
- 0) Exit

1

Enter the value for arr[0][0]: 1

Enter the value for arr[0][1]: 2

Enter the value for arr[1][0]: 3

Enter the value for arr[1][1]: 4

-----  
The determinant of the matrix is: -2

The matrix is linearly independent.  
-----

2

Enter the value for arr[0][0]: 1

Enter the value for arr[0][1]: 2

Enter the value for arr[0][2]: 3

Enter the value for arr[1][0]: 4

Enter the value for arr[1][1]: 5

Enter the value for arr[1][2]: 6

Enter the value for arr[2][0]: 7

Enter the value for arr[2][1]: 8

Enter the value for arr[2][2]: 9

-----  
The determinant of the 3x3 matrix is: 0

The matrix is linearly dependent.  
-----

3

Enter the size of the matrix (2 for 2x2, 3 for 3x3): 2

Enter values for first matrix:

Enter the value for arr[0][0]: 1

Enter the value for arr[0][1]: 2

Enter the value for arr[1][0]: 3

Enter the value for arr[1][1]: 4

Enter values for second matrix:

Enter the value for arr[0][0]: 5

Enter the value for arr[0][1]: 6

Enter the value for arr[1][0]: 8

Enter the value for arr[1][1]: 7

The result of matrix addition is:

6 8

11 11  
█



4

Enter the size of the matrix (2 for 2x2, 3 for 3x3): 3

Enter values for first matrix:

Enter the value for arr[0][0]: 1

Enter the value for arr[0][1]: 2

Enter the value for arr[0][2]: 3

Enter the value for arr[1][0]: 4

Enter the value for arr[1][1]: 7

Enter the value for arr[1][2]: 5

Enter the value for arr[2][0]: 3

Enter the value for arr[2][1]: 3

Enter the value for arr[2][2]: 2

Enter values for second matrix:

Enter the value for arr[0][0]: 6

Enter the value for arr[0][1]: 2

Enter the value for arr[0][2]: 7

Enter the value for arr[1][0]: 4

Enter the value for arr[1][1]: 1

Enter the value for arr[1][2]: 5

Enter the value for arr[2][0]: 3

Enter the value for arr[2][1]: 1

Enter the value for arr[2][2]: 5

The result of matrix subtraction is:

-5 0 -4

0 6 0

0 2 -3



5

Enter the row size and column size of the matrix 1 : 3 2

Enter the row size and column size of the matrix 2 : 2 3

Enter values for first matrix:

Enter the value for arr[0][0]: 1

Enter the value for arr[0][1]: 2

Enter the value for arr[1][0]: 3

Enter the value for arr[1][1]: 45

Enter the value for arr[2][0]: 4

Enter the value for arr[2][1]: 2

Enter values for second matrix:

Enter the value for arr[0][0]: 2

Enter the value for arr[0][1]: 4

Enter the value for arr[0][2]: 5

Enter the value for arr[1][0]: 2

Enter the value for arr[1][1]: 5

Enter the value for arr[1][2]: 1

The result of matrix multiplication is:

6 14 7

96 237 60

12 26 22



```
6
Enter the size of the matrix (2 for 2x2, 3 for 3x3): 3
Enter the value for arr[0][0]: 1
Enter the value for arr[0][1]: 2
Enter the value for arr[0][2]: 3
Enter the value for arr[1][0]: 4
Enter the value for arr[1][1]: 5
Enter the value for arr[1][2]: 7
Enter the value for arr[2][0]: 4
Enter the value for arr[2][1]: 2
Enter the value for arr[2][2]: 5
```

The result of transpose of matrix is:

```
1 4 4
2 5 2
3 7 5
```

```
█
```

```
7
Enter the row size and column size of the matrix : 3 3
Enter values for the matrix:
Enter the value for arr[0][0]: 1
Enter the value for arr[0][1]: 2
Enter the value for arr[0][2]: 3
Enter the value for arr[1][0]: 52
Enter the value for arr[1][1]: 1
Enter the value for arr[1][2]: 24
Enter the value for arr[2][0]: 1
Enter the value for arr[2][1]: 23
Enter the value for arr[2][2]: 5
Enter power for calculation: 2
```

The result of matrix raised to power 2 is:

```
108 73 66
128 657 300
1202 140 580
```

```
█
```

8

Entered Invalid Choice!

0

Exited calculator

PS C:\Users\j\_siv\OneDrive\Desktop\C\SEM-1(Mini project)> |

# Any Publication targeted

**YES / NO**