

# Matrix Calculator Project in C Programming

Explore the innovative Matrix Calculator Project built using C Programming, showcasing its functions, applications, and educational benefits.

Presenters:

**Sarvesh Sivasankaran**  
**Sam Devaraja J**

# Matrix Calculator Project Overview

Key Features of the Matrix Calculator



## Comprehensive Matrix Calculator

The project is a complete solution for matrix calculations using the C programming language.



## Supports 2x2 and 3x3 Matrices

Designed specifically to handle operations for both 2x2 and 3x3 matrices efficiently.



## User-Friendly Interface

The calculator features a simple interface for easy interaction and input of matrix values.



## Multiple Operations Available

It can perform various operations including addition, subtraction, multiplication, and determinant calculations.



## Educational Tool

This project serves as an educational tool for students learning about matrices and their applications.



## Open Source Project

The code is open source, allowing users to modify and enhance the calculator according to their needs.



## Error Handling Mechanisms

Includes robust error handling to ensure accurate calculations and user input validation.

# Matrix Calculator

## Project Overview

Enhancing Programming Skills in C



### Project Purpose

The project aims to develop a user-friendly console-based matrix calculator.



### Key Functionalities

It will perform various matrix operations such as Determinants, addition, subtraction, multiplication, Power and inversion.



### Educational Goals

The project enhances understanding of matrix mathematics and improves programming skills in C.

# Matrix Operations Tool Features

Key Functionalities and Educational Benefits

## Determinant Calculation and More

The tool allows for calculating determinants, performing matrix addition, subtraction, multiplication, transposition, and power computations.

## Educational Tool

This tool serves as a practical educational resource, enhancing understanding of essential matrix operations.

## User-Friendly Interface

An interactive, menu-driven interface enables users to easily select and perform their desired matrix operations.

# Matrix Operations Overview

Key Functionalities of Matrix Operations

## Determinant Calculation

Calculates determinants for 2x2 and 3x3 matrices, essential for linear algebra applications.

## Power Computation

Allows raising square matrices to a user-defined power, useful for advanced mathematical modeling.



## Matrix Operations

Supports addition, subtraction, multiplication, and transposition of matrices, enhancing computational flexibility.

## Modular Programming

Implements each operation as a distinct function, promoting code clarity and reusability in programming.

# Understanding Determinants in Matrices

Fundamental Concepts in Linear Algebra

## 2x2 Determinant Calculation

Calculated using the formula:  $\det = a*d - b*c$ , providing a straightforward method for evaluation.



## 3x3 Determinant Methodology

Utilizes cofactor expansion for calculation, which involves a more complex but systematic approach.



## Importance of Determinants

Essential for determining linear independence and assessing matrix properties such as invertibility.



# OUTPUT:

-----"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.  
-----



# Matrix Addition Function

Overview of the matrixAddition() Function

## Function Definition

The 'matrixAddition()' function executes element-wise addition for both 2x2 and 3x3 matrices.

## Matrix B Example

Matrix B is shown as: 5, 6; 7, 8. This is the second operand for the addition operation.

## Element-wise Addition Explained

Each corresponding element in the matrices is summed to produce the resultant matrix.

## Enhancing Computational Efficiency

Utilizing matrix addition can significantly enhance computational efficiency in algorithms.

## Matrix A Example

Matrix A is represented as: 1, 2; 3, 4. It serves as the first operand in addition.

## Result of Addition

The result of adding Matrix A and Matrix B yields: 6, 8; 10, 12.

## Applications of Matrix Addition

Matrix addition is fundamental in various fields including computer graphics and data analysis.

## Visualizing Matrix Operations

Visual aids help in understanding the process of matrix addition and its outcomes.

```
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
```



# Matrix Subtraction Function Overview

Element-wise Subtraction of Matrices

## Function Overview

The `matrixSubtraction()` function enables element-wise subtraction between two matrices, enhancing computational efficiency.

## How It Works

This function subtracts corresponding elements from two matrices of the same dimensions, yielding a new matrix.

## Example Matrices

Consider Matrix A: 5, 6, 7, 8 and Matrix B: 1, 2, 3, 4.

## Result of Subtraction

The subtraction results in: 4, 4, 4, 4, demonstrating the element-wise difference.

```
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
```



# Matrix Multiplication Function Overview

Key Details and Example of Matrix Multiplication

## Function Overview

The function `matrixMultiplication()` multiplies two matrices while checking for compatibility.

## Matrix A Details

Matrix A is a 2x3 matrix with values: 1, 2, 3; 4, 5, 6.

## Matrix B Details

Matrix B is a 3x2 matrix with values: 7, 8; 9, 10; 11, 12.

## Resulting Matrix

The resulting matrix from the multiplication is: 58, 64; 139, 154.



5

Enter the row size and column size of the matrix 1  
Enter the row size and column size of the matrix 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]: 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





# Matrix Transposition Function

Understanding the transpose() Function



## Understanding the transpose() function

The function transpose() computes the transpose of a 2x2 or 3x3 matrix by swapping rows and columns.



## Original Matrix Example

Consider the original 2x2 matrix:

1	2
3	4



## Resulting Transposed Matrix

The transposed matrix is:

1	3
2	4

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



# Matrix Power Function Explained

Exploring how to raise matrices to powers

## Functionality of power()

The `power()` function efficiently raises a square matrix to a specified power using repeated multiplication.



## Matrix Example

Consider the matrix  $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$  as an example for power calculation.



## Result of Squaring

Raising the matrix to the power of 2 results in  $\begin{pmatrix} 7 & 10 \\ 15 & 22 \end{pmatrix}$ , showcasing the function's capability.



```
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
```

# Invalid Choice Handling:

```
-----"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  
6) Transpose of a matrix (2x2 or 3x3)  
7) Power of a matrix (2x2 or 3x3)  
0) Exit  
  
8  
Entered Invalid Choice!  
0  
Exited calculator
```

# Modular Programming: Structure and Functionality

Understanding the Benefits of Modular Programming



## Enhances Code Readability

Modular programming organizes code into distinct functions, making it easier to read and understand.



## Improves Reusability

Functions can be reused across different projects, reducing redundancy and saving time.



## Facilitates Maintainability

Modular design simplifies updates and maintenance, allowing developers to focus on individual components.



## Easier Debugging

Isolated functions allow for more straightforward debugging, making it easier to identify and fix issues.



## Streamlined Updates

Changes can be made to specific functions without impacting the entire codebase, enhancing efficiency.

# Program Error Handling Features

Enhancing User Experience and Stability

## 01 Error Handling Mechanisms

The program employs robust error handling to verify user inputs, ensuring validity and preventing errors.

## 02 User Guidance for Input

Users receive clear guidance on selecting valid matrix dimensions and operations, minimizing runtime errors.

## 03 Enhanced User Experience

By preventing input errors, the program enhances overall user experience, making it more intuitive and accessible.

## 04 Stability of the Program

Effective error handling contributes significantly to the stability of the program, reducing crashes and unexpected behavior.

# Literature Survey: Existing Applications

A Lightweight Approach for Educational Use



## Advanced Functionalities in Applications

01

Applications like MATLAB and online matrix calculators offer complex features for matrix computations.

## Focus on Fundamental Operations

02

This project differentiates itself by emphasizing basic matrix operations, fostering a foundational understanding.



# Educational Command-Line Project Overview

Key Features and Accessibility

## Designed for Educational Purposes

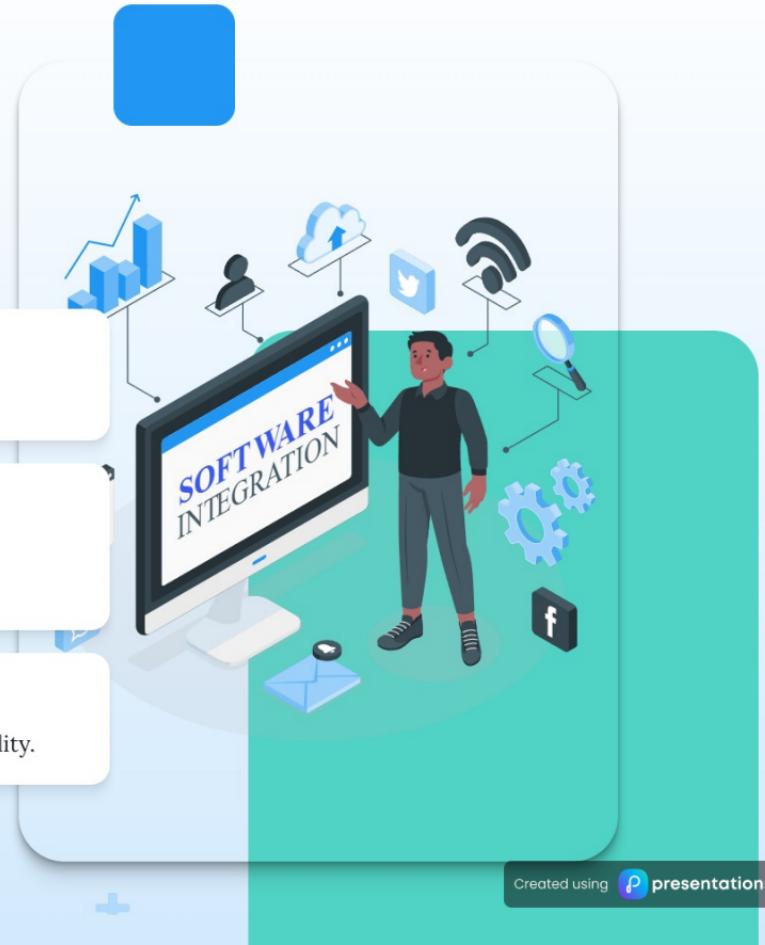
This project is tailored for beginner programmers, making it an ideal learning tool.

## Command-Line Interface Utilization

The command-line interface simplifies the programming process compared to GUI applications.

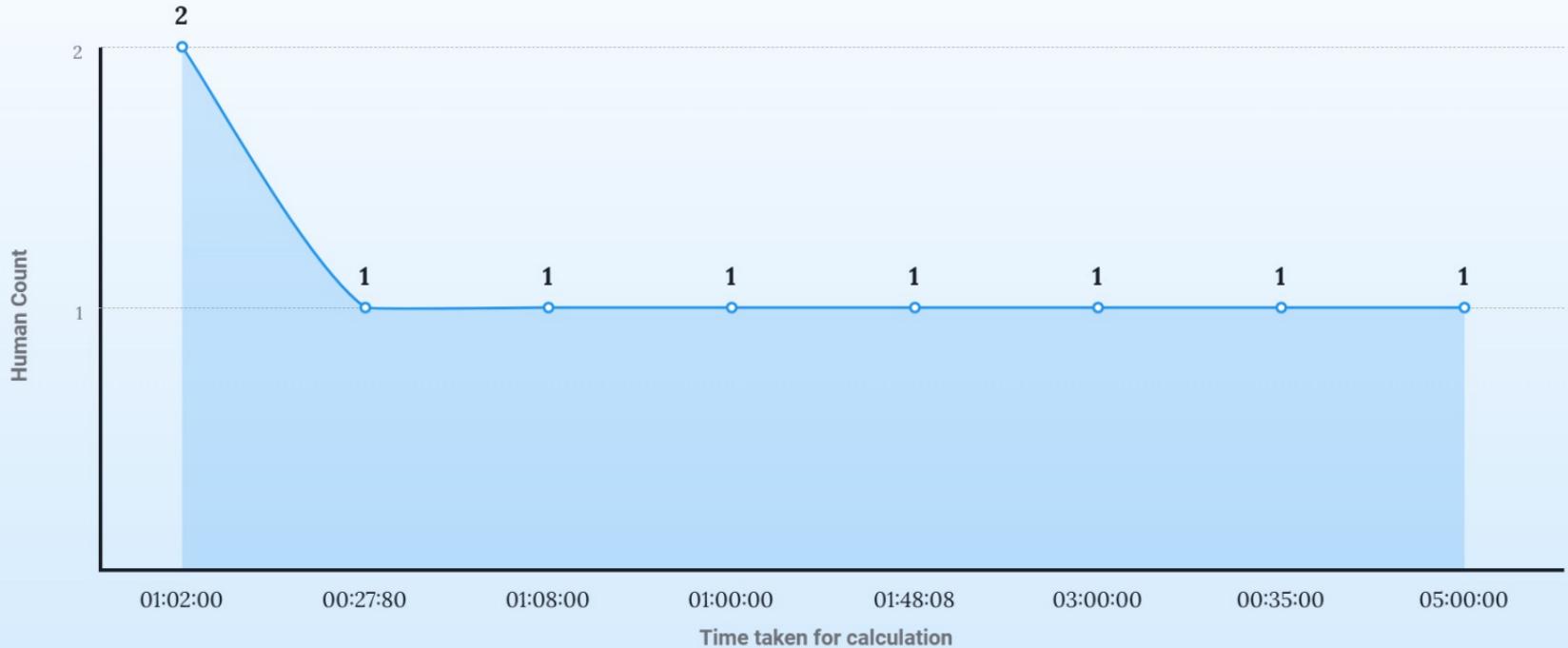
## Wide System Compatibility

It runs on basic systems equipped with a standard C compiler, enhancing accessibility.



# Literature Survey:

Your determinant calculation Timing without calculator(minutes)



# Survey Result:

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





# Matrix Calculator Project Overview

Enhancing Understanding of Matrix Operations

## ■ Educational tool for matrix operations

The Matrix Calculator project serves as an educational tool, helping users grasp matrix operations effectively.

## ■ Focus on programming concepts and modular design in C

The project emphasizes fundamental programming concepts, showcasing modular design principles in C programming.

# Thank You

for your attention! Your feedback and suggestions are greatly appreciated as we continue to learn and improve through this project.