Sanjay Ghodawat University Kolhapur

Established as a State Private University under Govt. of Maharashtra Act No. XL dated 3rd May 2017

Empowering Lives Globally !

PROJECT REPORT

A report submitted in partial fulfillment of the requirements for the

Project

School of Computer Science & Engineering

By

Veetarag V Patil
PRN No:22SC114281031 Roll No: 28

Program: BTech Class: FY BTech (Div. A)

Under Supervision of

Mr. Sanket S. Prabhu

Academic Year: 2022-2023

Sanjay Ghodawat University Kolhapur

Established as a State Private University under Govt. of Maharashtra Act No. XL dated 3rd May 2017

Empowering Lives Globally !

School of Computer Science & Engineering

CERTIFICATE

This is to certify that the "**Project Report**"

On

MATRIX CALCULATOR

submitted by

Veetarag v Patil

PRN No:22SC114281031 Roll No: 28

Program: BTech Class: FY BTech(Div A)

is work done by him/her and submitted during the 2022 – 2023 academic year, in partial fulfillment of the **Project.**

Sanjay Ghodawat University, Kolhapur

Mr. Sanket Prabhu	**Ms. Deepika Patil**	**Dr. B. Suresh Kumar**	
Project Guide	**PBL Co-ordinator**	**Head, SOCSE**	**External**

DECLARATION

I the undersigned solemnly declare that the report of the project work entitled **"Project Name"** which is carried out under the supervision of**Mr. Sanket S. Prabhu** I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the project report does not contain any part of any work which has been submitted for the award of any other degree/diploma/certificate in this University or any other University.

Student Name:

Veetarag v Patil
PRN No:22SC114281031 Roll No: 28

Class: FY BTech (Div A)

ACKNOWLEDGMENT

First, I would like to thank my Head of the School **Dr. B. Suresh Kumar** for constructive criticism throughout my project. I would like to thank PBL coordinator **Ms. Deepika Patil** and Department Project Guide **Mr. Sanket S Prabhu** for support and advices to get and complete internship in above said organization. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals. I am extremely grateful to my department staff members and friends who helped me in successful completion of this project.

ABSTRACT

- This program is written in C programming language and performs matrix addition, subtraction, and multiplication operations.
- The program takes input matrices and performs the desired operation. The resulting matrix is then displayed.
- The program is implemented using C programming constructs and libraries. The program also checks for the compatibility of matrices before performing the operation.
- It's a simple yet efficient program that can be used as a building block for more complex matrix-based programs.

TABLE OF CONTENT'S

Introduction

- **Introduction**

A matrix is a rectangular array of numbers or symbols arranged in rows and columns. There are different types of matrices like row matrix, column matrix, horizontal matrix, vertical matrix, square matrix, diagonal matrix, identity matrix, equal matrix, singular matrix, etc. The three basic matrix operations are **addition, subtraction, and multiplication**. The following section contains various C programs on matrix operations, matrix types, matrix diagonals, sparse matrix, invertible matrix, and adjacency matrix.

Each sample program on the matrix includes a program description, C code, and program output. All examples have been compiled and tested on Windows
and Linux systems.

- **Problem Definition**

1. **Input:** The input to the program would typically include the matrices to be added, subtracted, or multiplied, as well as any additional information needed for the specific problem, such as the dimensions of the matrices.

2. **Output**: The output of the program would typically include the resulting matrix from the addition, subtraction, or multiplication operation, as well as any additional information needed for the specific problem.

3. **Constraints:** The problem definition would typically include any constraints on the input matrices or the operation, such as the dimensions of the matrices or the type of numbers they contain.

Scope

- Basic matrix operations: Addition, subtraction and multiplication of matrices.
- Linear algebra: Matrix addition, subtraction, and multiplication are fundamental operations in linear algebra, and are used in many algorithms and computations in this field.
- Computer graphics: Matrix addition, subtraction, and multiplication are used for performing transformations on objects in computer graphics, such as translation, rotation, and scaling.
- Scientific computing: These operations are used in various scientific computing applications, such as solving partial differential equations and simulating physical systems.
- Machine learning: Matrix addition, subtraction, and multiplication are used in many machine learning algorithms, such as linear regression and neural networks.
- Robotics: Robotics heavily relies on Linear Algebra, these operations can be used for controlling the movement of robots and determining the position and orientation of robots in space.
- Overall, matrix addition, subtraction, and multiplication are fundamental operations in many fields, and are used in a wide range of applications. Implementing these operations in C programming can provide a foundation for building more advanced and specialized matrix-based programs and apply.

Problem Identification

1. Incorrect matrix dimensions: If the matrices being added, subtracted, or multiplied have different dimensions, the program will not work as expected.

2. Memory allocation issues: Improper memory allocation for the matrices can result in incorrect results or program crashes.

3. Incorrect mathematical operations: If the mathematical operations for matrix addition, subtraction, and multiplication are implemented incorrectly, the program will not produce the correct results.

4. Data type issues: Mixing data types such as using float values for one matrix and int values for another can result in unexpected results.

5. Input validation: Failing to validate user input can result in incorrect results or crashes.

6. Printing errors: Printing the results of the matrix operations without proper formatting can make it difficult to interpret the results.

Objectives

- This program will help student to solve matrix problem easily.
- C programming language supports matrix as a data type and offers more flexibility. And also it consumes less memory while processing.
- By storing values in a matrix rather than as individual variables, C program can access and perform operations on the data more efficiently.
- It is easier to extract information about object rotation, and also easy to manipulate in the C program**.**

System Requirements Specification

- **Software Requirement:**

  Dev C++

- **Hardware Requirement**

  Laptop

  Intel(R) Core(TM) i3-Processor

  RAM-1 GB Minimum

  Storage-100GB

Methodology

- **Algorithm**

Step 1: Start the Program.

Step 2: choice the calculator option

Step 3: Enter the row and column of the first (a) matrix.

Step 4: Enter the row and column of the second (b) matrix.

Step 5: Enter the elements of the first (a) matrix.

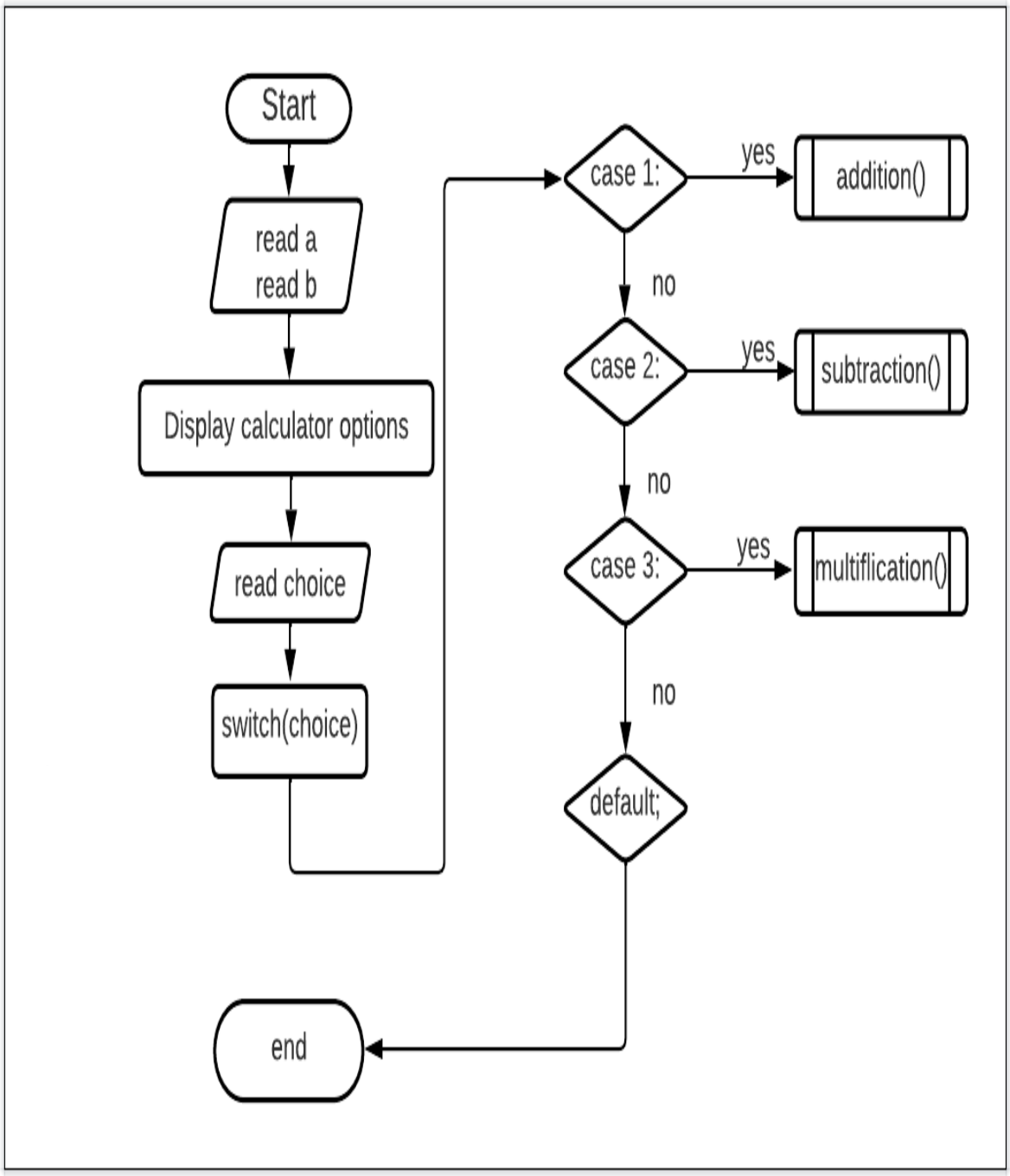
Step 6: Enter the elements of the second (b) matrix.

Step 7: process of calculation which you have selected

Step 8: printing final matrix

Step 9: End of program

- **Flow Diagram (Flow Chart):**



Implementation

This program prompts the user to enter the number of rows and columns for two matrices, A and B, then allows the user to enter the elements of each matrix. The program then adds the two matrices together and prints out the result.

Note that this is a simple example and you can add different operation like transpose, determinant, rank etc.

“Introduction of C Programming”

The native language of a computer is binary—ones and zeros—and all instructions and data must be provided to it in this form. Native binary code is called machine language. The earliest digital electronic computers were programmed directly in binary, typically via punched cards, plug-boards, or front-panel switches. Later, with the advent of terminals with keyboards and monitors, such programs were written as sequences of hexadecimal numbers, where each hexadecimal digit represents a four binary digit sequence. Developing correct programs in machine language is tedious and complex, and practical only for very small programs.

C is a general-purpose programming language, and is used for writing programs in many different domains, such as operating systems, numerical computing, graphical applications, etc. It is a small language, with just 32 keywords It provides ―high-level structured-programming constructs such as statement grouping, decision making, and looping, as well as ―low-level capabilities such as the ability to manipulate bytes and addresses. Since C is relatively small, it can be described in a small space, and learned quickly. A programmer can reasonably expect to know and understand and indeed regularly use the entire language

Advantages of learning C:

1) It is easy to understand :

   One of the main reasons why people choose C over other programming languages is its simplicity. C is a highly portable language as programs coded in it are far more fast and efficient. This makes learning C easier than any other programming language.

2) Presence of many Libraries:

   C Language provides lots of built-in functions which consist of system-generated functions and user-defined functions.

3) Easy to write:

   Another reason why C is so popular as an efficient language among programmers is that it allows them to create their own software without having to worry about syntax errors.

4) Low cost :

   If you want to build something from scratch, then C is definitely worth considering. Because of its simple structure, you won't spend too much time trying to figure out whether you've made a mistake or not when developing your program.

5) Fast execution speed:

   If you want to execute your application quickly, then C is probably the right choice for you. Since C uses fewer instructions, it executes faster than other programming languages such as Java, Ruby, PHP, etc.

Source Code:

```
#include<stdio.h>
int main()
{
int r,c,p,q,k,a[50][50],b[50][50],sum[50][50],i,j,cal;
printf("1.addition\n2.subtraction\n3.multiflication\nENTER YOUR CHOICE:");
scanf("%d",&cal);
printf("enter the number of rows and columns for: A(between 1 to 50):");
scanf("%d%d",&r,&c);
printf("enter the number of rows and columns for: B(between 1 to 50):");
scanf("%d%d",&p,&q);
printf("\nenter element of 1st matrix:\n");
for(i=0; i<r; ++i)
for(j=0; j<c; ++j)
{
printf("enter element a%d%d:",i+1,j+1);
scanf("%d", &a[i][j]);
}
printf("enter elements of 2nd matrix:\n");
for(i=0; i<p; ++i)
for(j=0; j<q; ++j)
{
printf("enter element b%d%d:",i+1,j+1);
scanf("%d",&b[i][j]);
}
switch(cal)
{
case 1:
//adding two matrices
for(i=0; i<r; ++i)
for(j=0; j<c; ++j)
{
sum[i][j]=a[i][j]+b[i][j];
}
//printing the result
printf("\n sum of two matrices:\n");
for(i=0; i<r; ++i)
for(j=0; j<c; ++j)
{
printf("%d\t",sum[i] [j]);
if(j==c-1)
{
printf("\n");
}
}
break;
case 2:
//subtraction two matrices
```

```
for(i=0; i<r; ++i)

for(j=0; j<c; ++j)
{
sum[i][j]=a[i][j]-b[i][j];
}
//printing the result
printf("\n subtraction of two matrices:\n");
for(i=0; i<r; ++i)
for(j=0; j<c; ++j)
{
printf("%d\t",sum[i][j]);
if(j==c-1)
{
printf("\n");
}
}
break;
case 3:
//multiplication of two matrices
for (i = 0; i < r; i++)
{
for (j = 0; j < q; j++)
{
for (k = 0; k < p; k++)
{
sum[i][j] += a[i][k] * b[k][j];
}
}
}
//printing the result
printf("The product of the two matrices is: \n");
for (i = 0; i < r; i++)
{
for (j = 0; j < q; j++)
{
printf("%d\t", sum[i][j]);
}
printf("\n");
}
break;
//for wrong input
default:
printf("you entered wrong input");
}
return 0;
}
```

Result

```
1.addition
2.subtraction
3.multiflication
ENTER YOUR CHOICE:2
enter the number of rows and columns for: A(between 1 to 50):3
3
enter the number of rows and columns for: B(between 1 to 50):3
3
```

```
enter element of 1st matrix:
enter element a11:1
enter element a12:2
enter element a13:3
enter element a21:4
enter element a22:5
enter element a23:6
enter element a31:7
enter element a32:8
enter element a33:9
enter elements of 2nd matrix:
enter element b11:0
enter element b12:1
enter element b13:2
enter element b21:3
enter element b22:4
enter element b23:5
enter element b31:6
enter element b32:7
enter element b33:8
subtraction of two matrices:
1   1   1
1   1   1
1   1   1
```

Conclusion & Future Scope

- **Future Scope:**

1. Optimization: With the advancement of computer hardware, the optimization of matrix operations in C programming is crucial for achieving faster results and reducing computational time.
2. Parallel computing: With the increasing demand for high-performance computing, parallel computing techniques can be applied to matrix operations to improve the computational efficiency.
3. Machine learning applications: Matrix operations play a crucial role in many machine learning algorithms, such as neural networks and support vector machines. As machine learning continues to advance, the importance of matrix operations in this field is expected to grow.
4. Graphic processing: Matrix operations are widely used in computer graphics for transforming and manipulating 2D and 3D objects. With the increasing demand for high-quality graphics and animations, the scope for matrix operations in computer graphics is expected to grow.
5. Big data processing: Matrix operations can be used for processing large datasets in areas such as data mining and scientific computing. With the growth of big data, the importance of efficient matrix operations in this field is expected to increase

- **Conclusion:**

- In conclusion, matrix addition, subtraction, and multiplication are essential concepts in linear algebra and computer programming.

- These operations are widely used in various fields such as computer graphics, computer vision, machine learning, and more.

- The implementation of matrix operations in C programming involves reading the elements of the matrices, performing the mathematical operations, and displaying the result.

- The future scope of matrix operations in C programming is vast, with potential developments in areas such as optimization, parallel computing, machine learning applications, graphic processing, and big data processing.

- Overall, matrix operations are fundamental to many areas of computer science and engineering and their importance is expected to continue to grow in the future.

References

Websites:

1. WWW.javatpoint.com
2. WWW.easycalculation.com

Books:

1. **computer fundamentals and programing in c**
   - **Reema thareja**
2. **C,the complete reference**
   - **Herbert schildt**