Project on

Scientific Calculator

Course title: Data Structure Lab
Course code: CSE 134

Submitted by,

Abida Jannat Chy[232-115-328**]**
Nehala Sultana[232-115-333**]**
Habiba Rahman Tahsin[232-115-348**]**
Batch: 59th
Section: I
Department of
C.S.E.

Submitted to,

Rishad Amin Pulok
Lecturer
Department of C.S.E.

Submission date :12.10.2024

Scientific Calculator

Introduction:

This project is a scientific calculator program written in C, designed to perform a wide variety of mathematical operations. It enables users to perform basic arithmetic, trigonometric calculations, logarithmic and exponential functions, matrix operations, factorials, and solve linear equations. It also supports functions for power, square root, cube root, and factorial calculations.Theprogram offers a user-friendly menu-driven interface that allows users to select specific operations repeatedly until they decide to exit. Also Designed for students, teachers, and professionals, it goes beyond basic calculations, offering advanced functions that are commonly used in physics,chemistry,engineering and other scientific fields.The calculator demonstrates core programming techniques, including function-based modular design, error handling, and control flow.

Background Study:

Choosing to build a scientific calculator is a valuable learning project because it provides a comprehensive introduction to programming concepts, enhances mathematical skills, and lays a strong foundation for more advanced projects.The concept of the science calculator originated from a desire to create a tool that simplifies complex calculations and aids learning.

The scientific calculators implemented in various programming languages (Python, Java, C,C++ etc.) that are similar to our project. We got inspired from this link and implemented it in our project and added some thingsfrom this link [http://github.com.](http://github.com)

Alsosome more sources we collect code and modified it by own. We know it has various features. We have tried to implement many of the features by using code. But still, there are lot of features that can be implemented. we can consider it one of the lacking.

How does the program work:

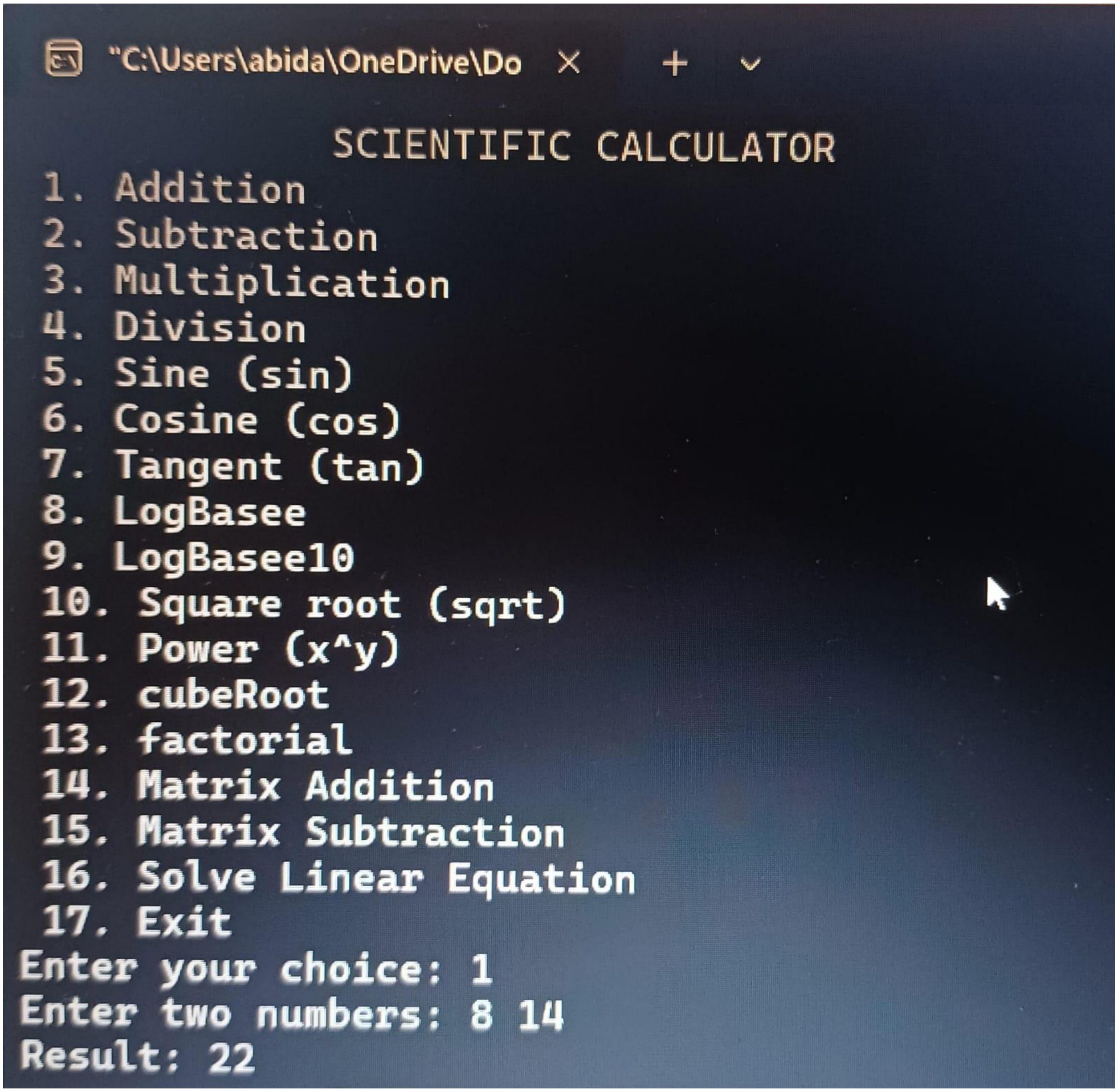
This scientific calculator program is written in C and provides a range of mathematical functions.The program is designed with modularity in mind, organizing each operation into a separate function for better readability and maintenance.The program includes functions for basic arithmetic (addition, subtraction, multiplication, division), trigonometric operations (sine, cosine, tangent), logarithmic calculations (natural and base-10 logs), power and root calculations, factorials, and matrix operations (addition and subtraction). It also includes a function to solve systems of linear equations using determinants. Each function performs checks to ensure that invalid inputs.

The`main()` function which displays available operations and prompts the user to select an option by entering a number. Based on the user's choice, the program calls the relevant function to perform the calculation. The program runs continuously in a loop, displaying the menu again after each operation, until the user chooses the exit option, which stops the program.

Features:

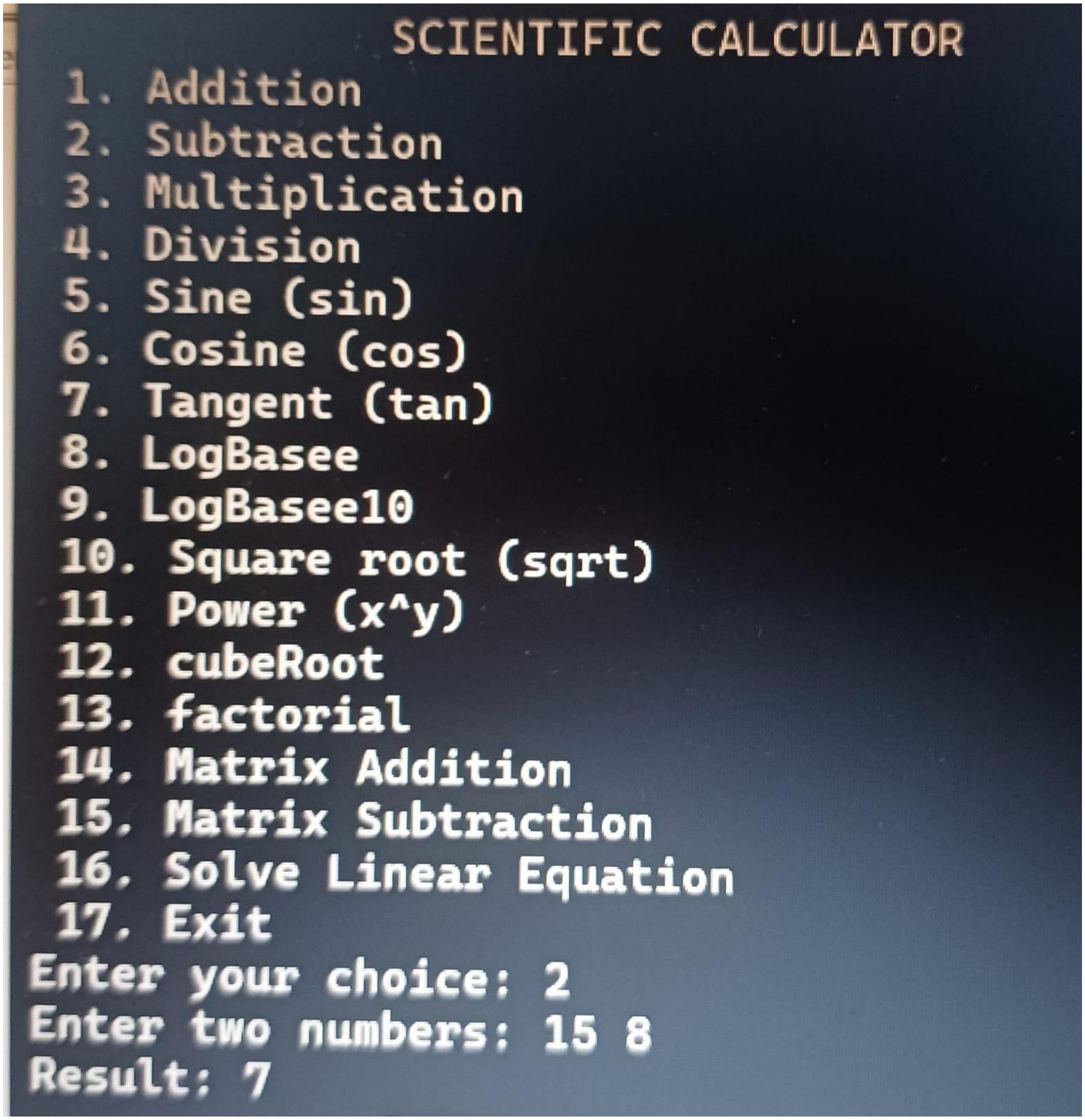
Here are some common features that we implement in our project.

1.Addition:



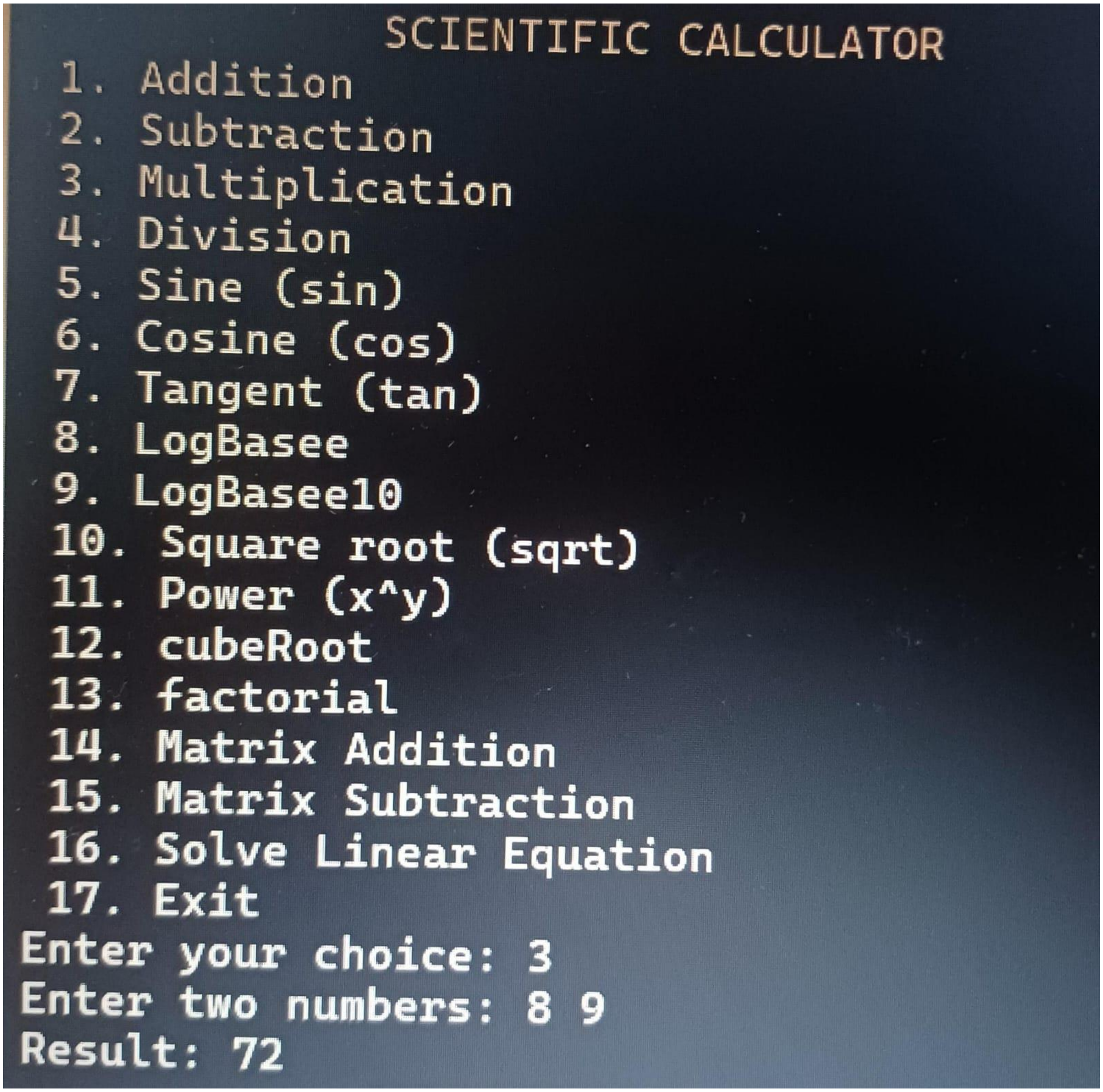
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 1
Enter two numbers: 8 14
Result: 22
```

2.Substruction:



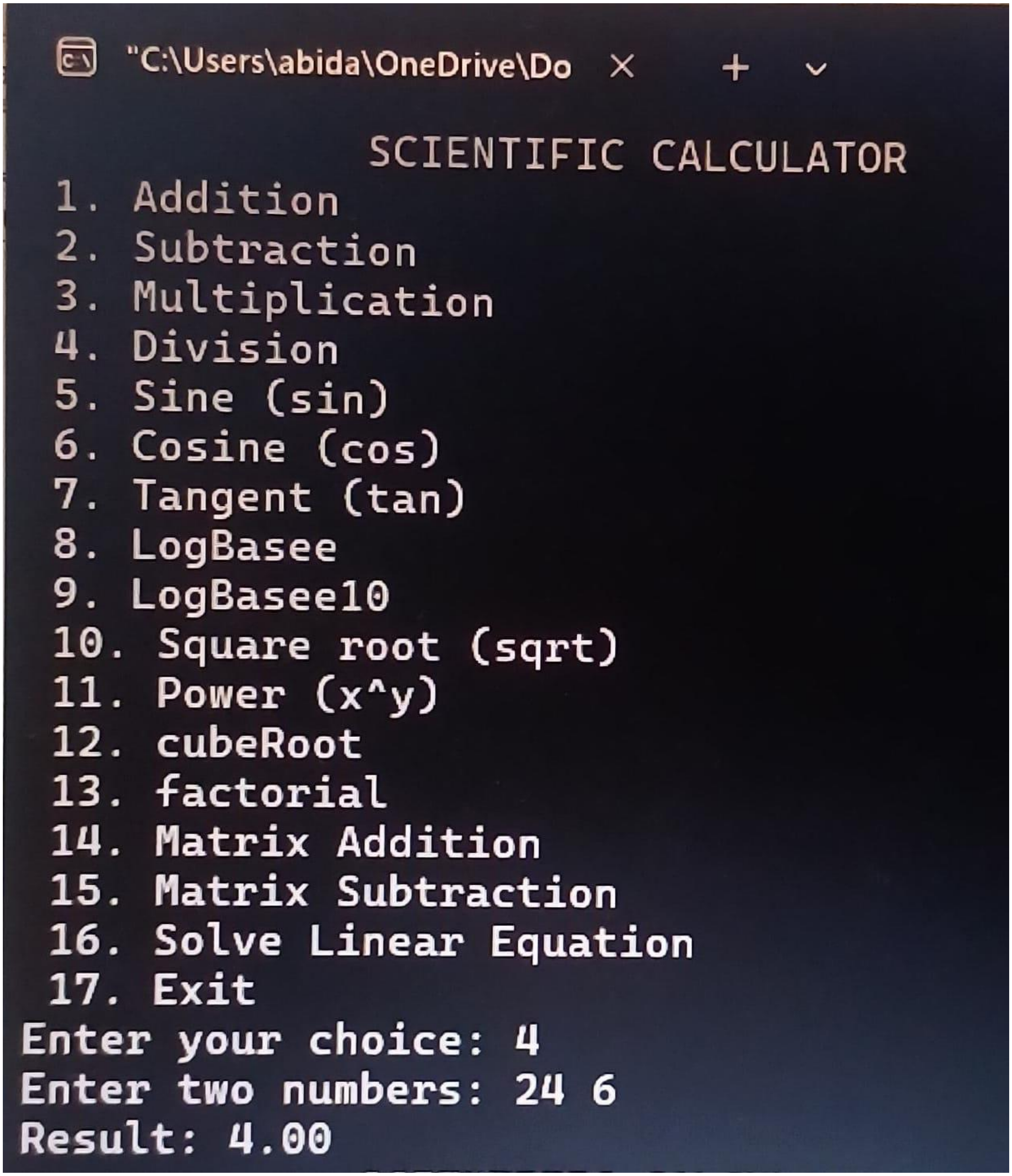
```
                  SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 2
Enter two numbers: 15 8
Result: 7
```

3.Multiplication:



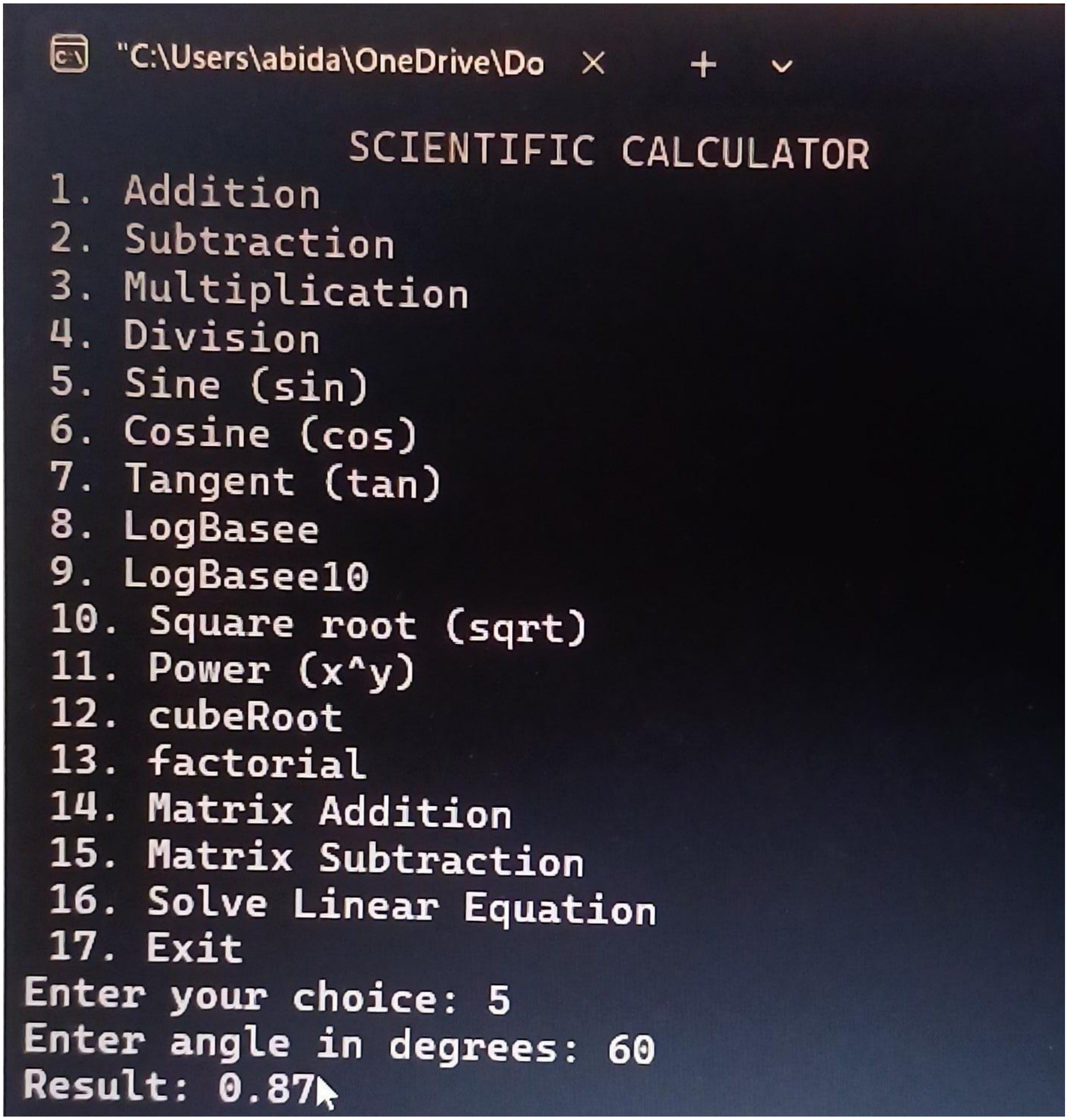
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 3
Enter two numbers: 8 9
Result: 72
```

4.Division:



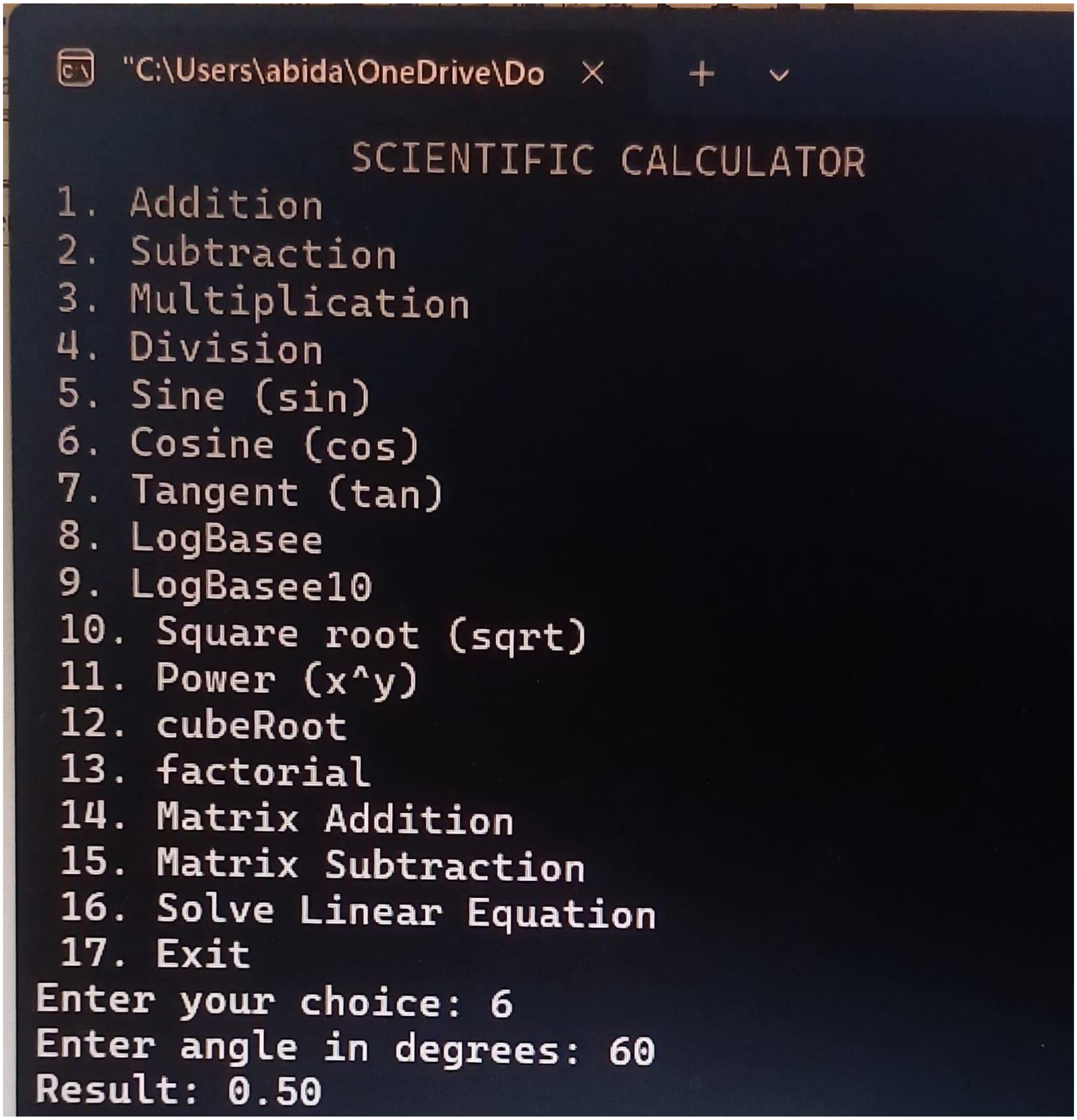
```
              SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 4
Enter two numbers: 24 6
Result: 4.00
```

5.Sine:



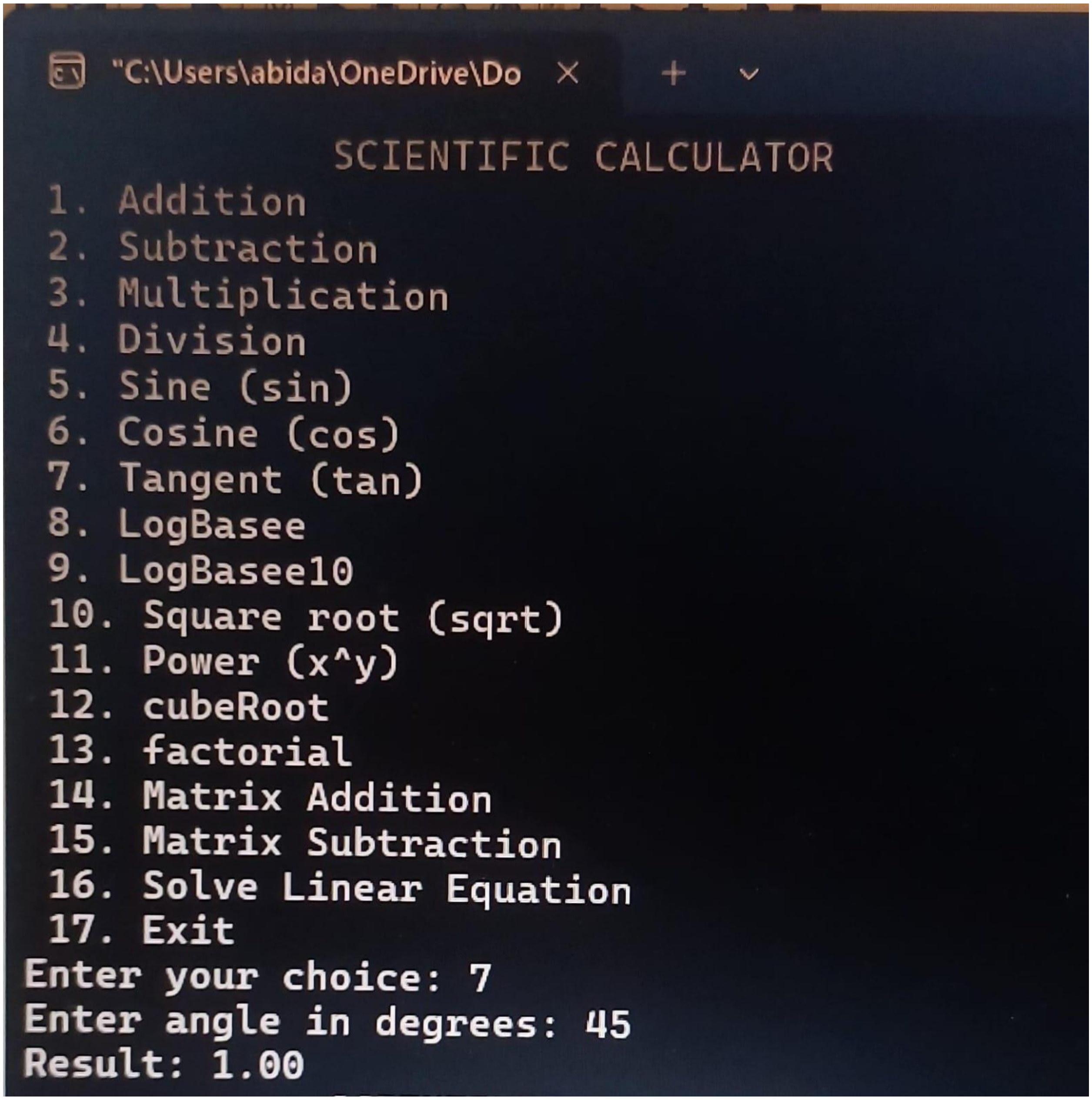
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 5
Enter angle in degrees: 60
Result: 0.87
```

6.Cosine:



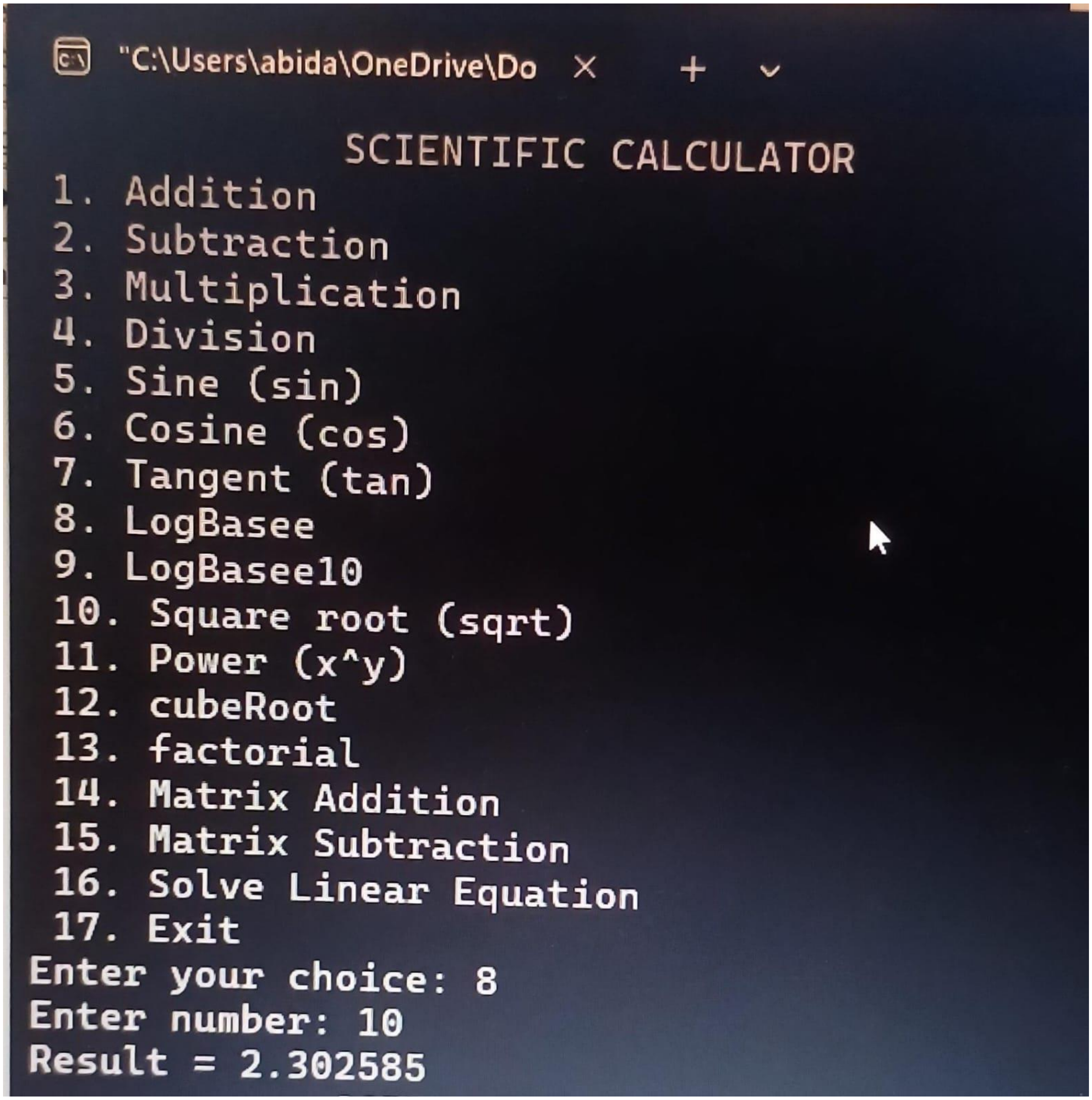
```
                    SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 6
Enter angle in degrees: 60
Result: 0.50
```

7.Tangent:



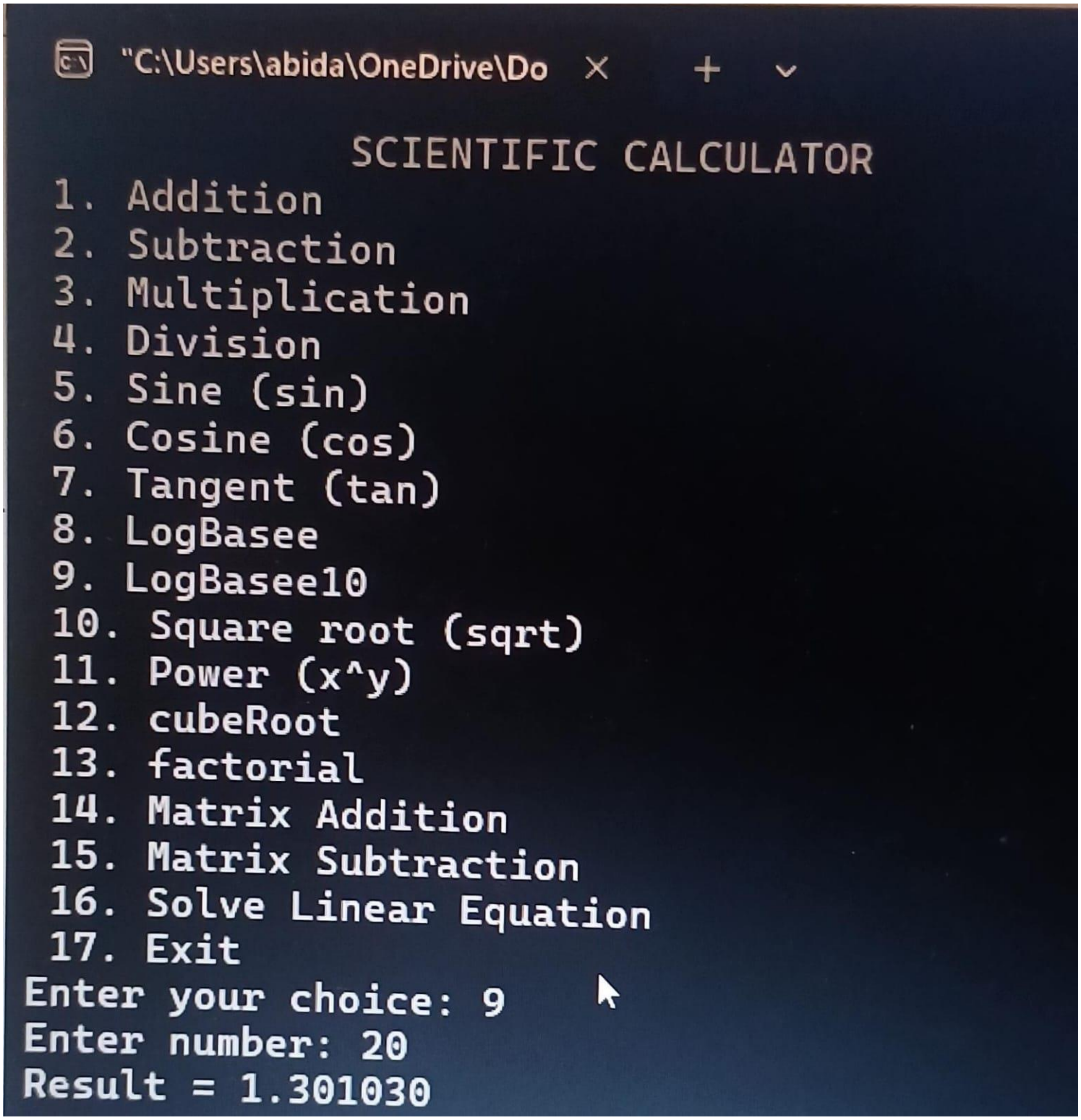
```
                 SCIENTIFIC CALCULATOR
  1. Addition
  2. Subtraction
  3. Multiplication
  4. Division
  5. Sine (sin)
  6. Cosine (cos)
  7. Tangent (tan)
  8. LogBasee
  9. LogBasee10
  10. Square root (sqrt)
  11. Power (x^y)
  12. cubeRoot
  13. factorial
  14. Matrix Addition
  15. Matrix Subtraction
  16. Solve Linear Equation
  17. Exit
Enter your choice: 7
Enter angle in degrees: 45
Result: 1.00
```

8.Logbasee:



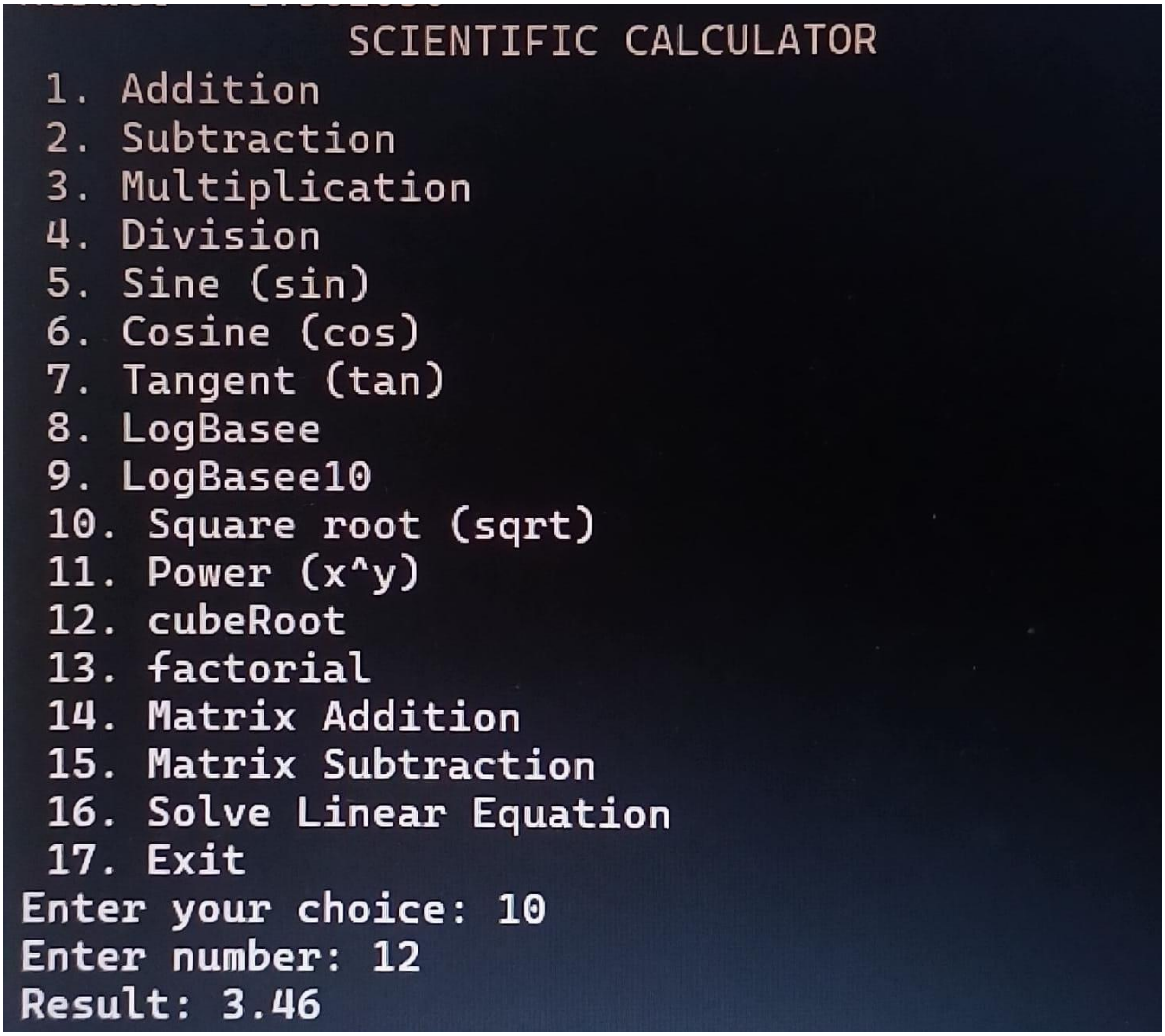
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 8
Enter number: 10
Result = 2.302585
```

9.LogBasee10:



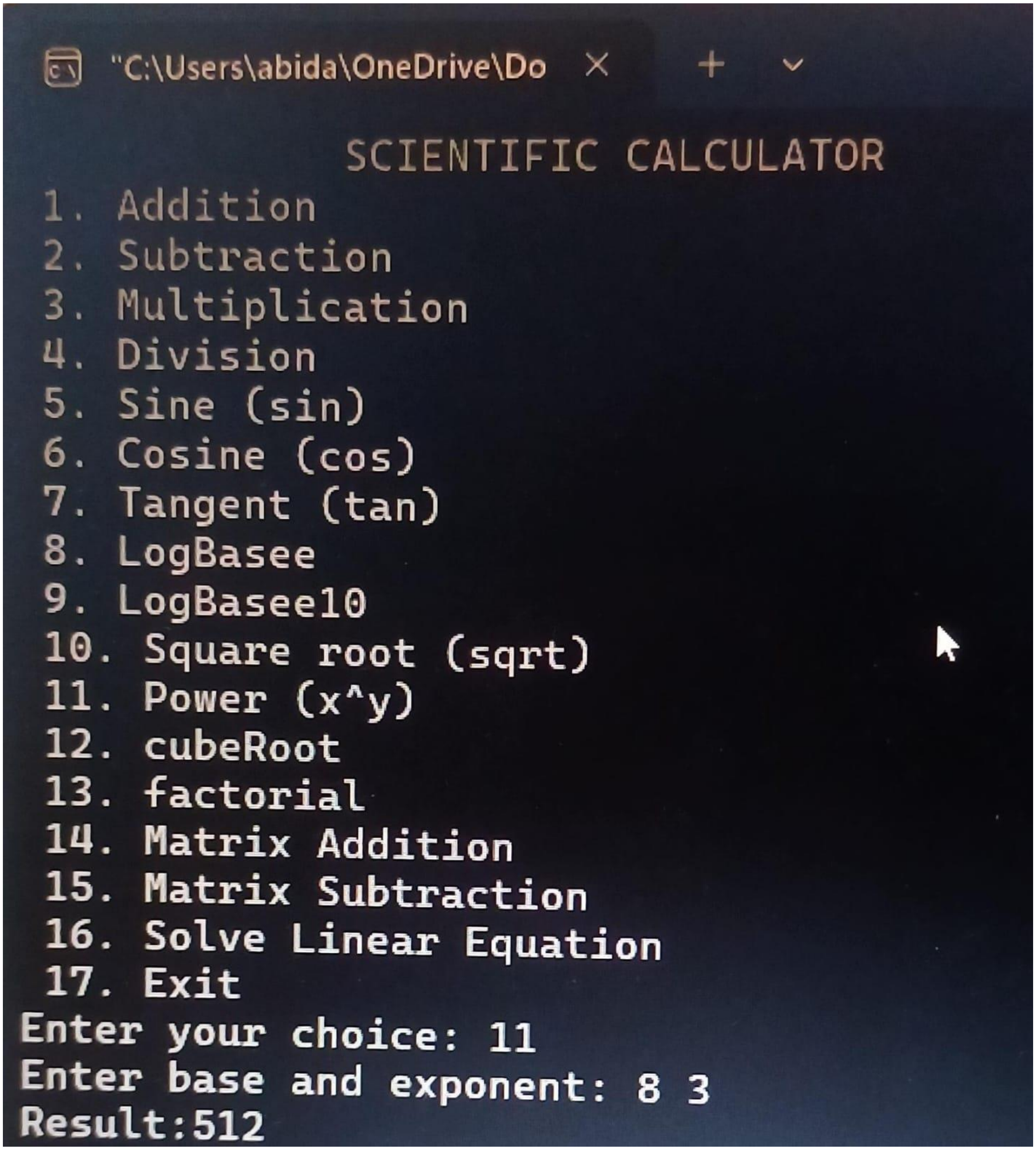
```
                    SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 9
Enter number: 20
Result = 1.301030
```

10.Squrare root:



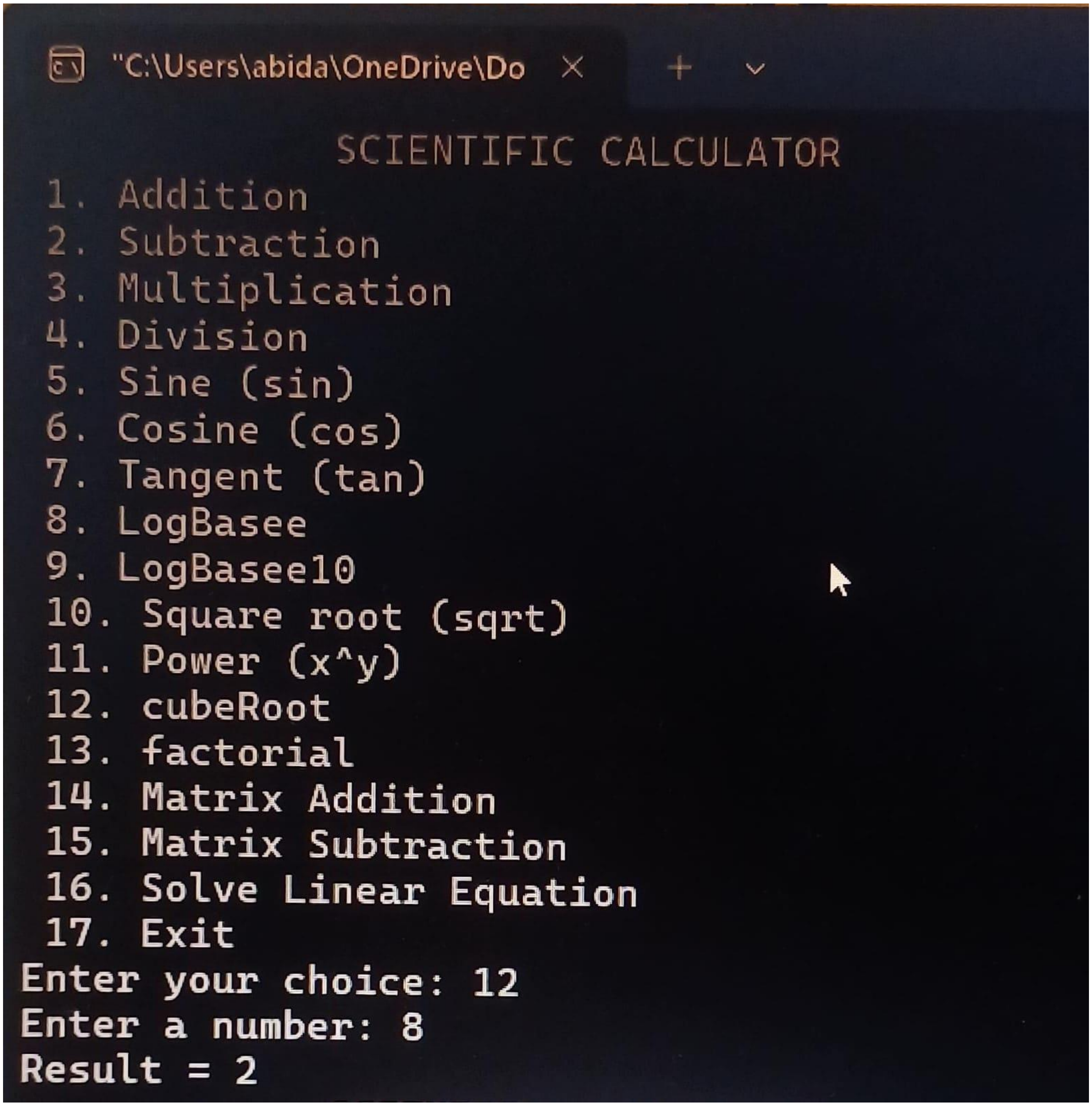
```
                    SCIENTIFIC CALCULATOR
  1. Addition
  2. Subtraction
  3. Multiplication
  4. Division
  5. Sine (sin)
  6. Cosine (cos)
  7. Tangent (tan)
  8. LogBasee
  9. LogBasee10
  10. Square root (sqrt)
  11. Power (x^y)
  12. cubeRoot
  13. factorial
  14. Matrix Addition
  15. Matrix Subtraction
  16. Solve Linear Equation
  17. Exit
Enter your choice: 10
Enter number: 12
Result: 3.46
```

11.Power:



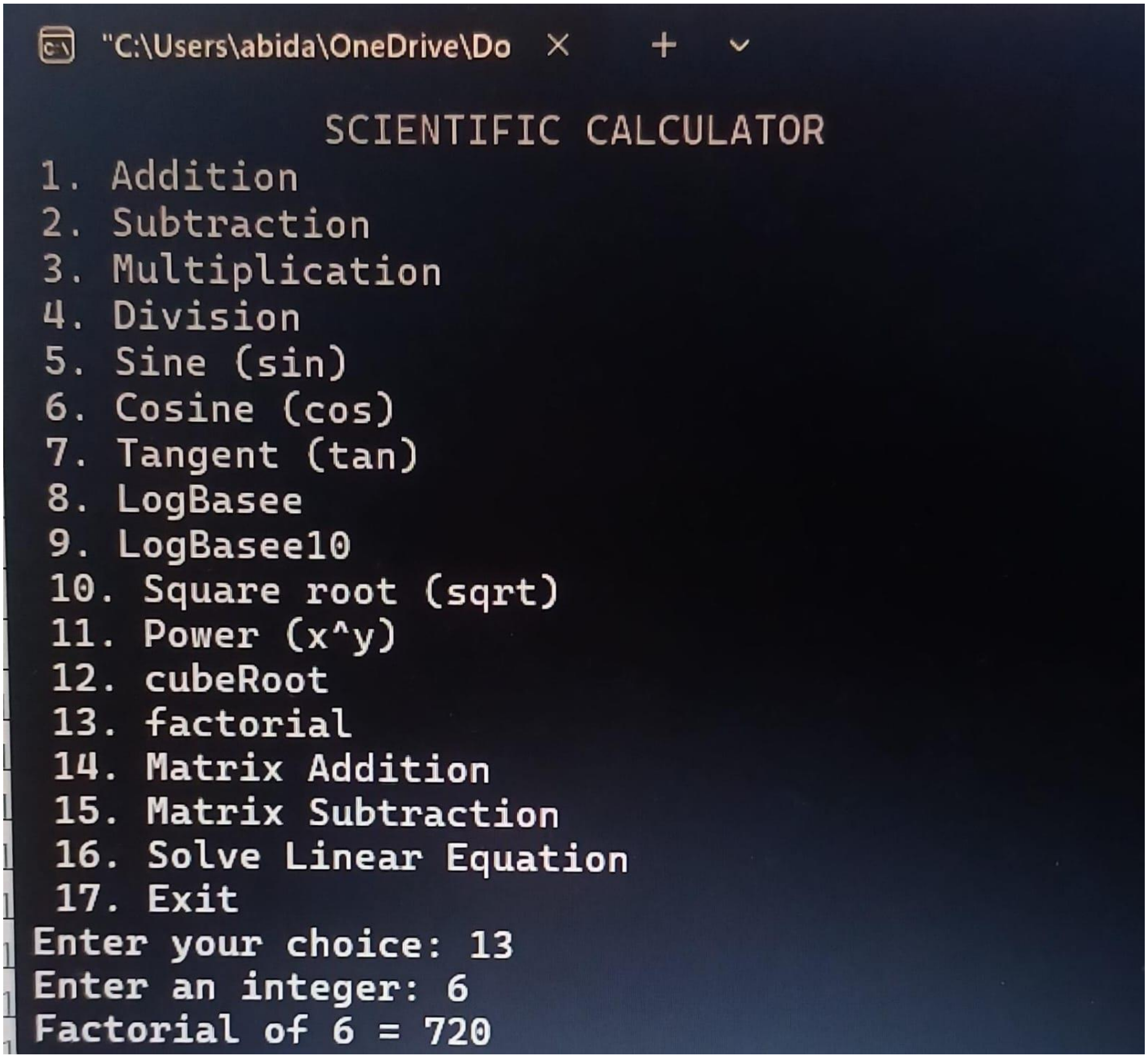
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 11
Enter base and exponent: 8 3
Result:512
```

12.cubeRoot:



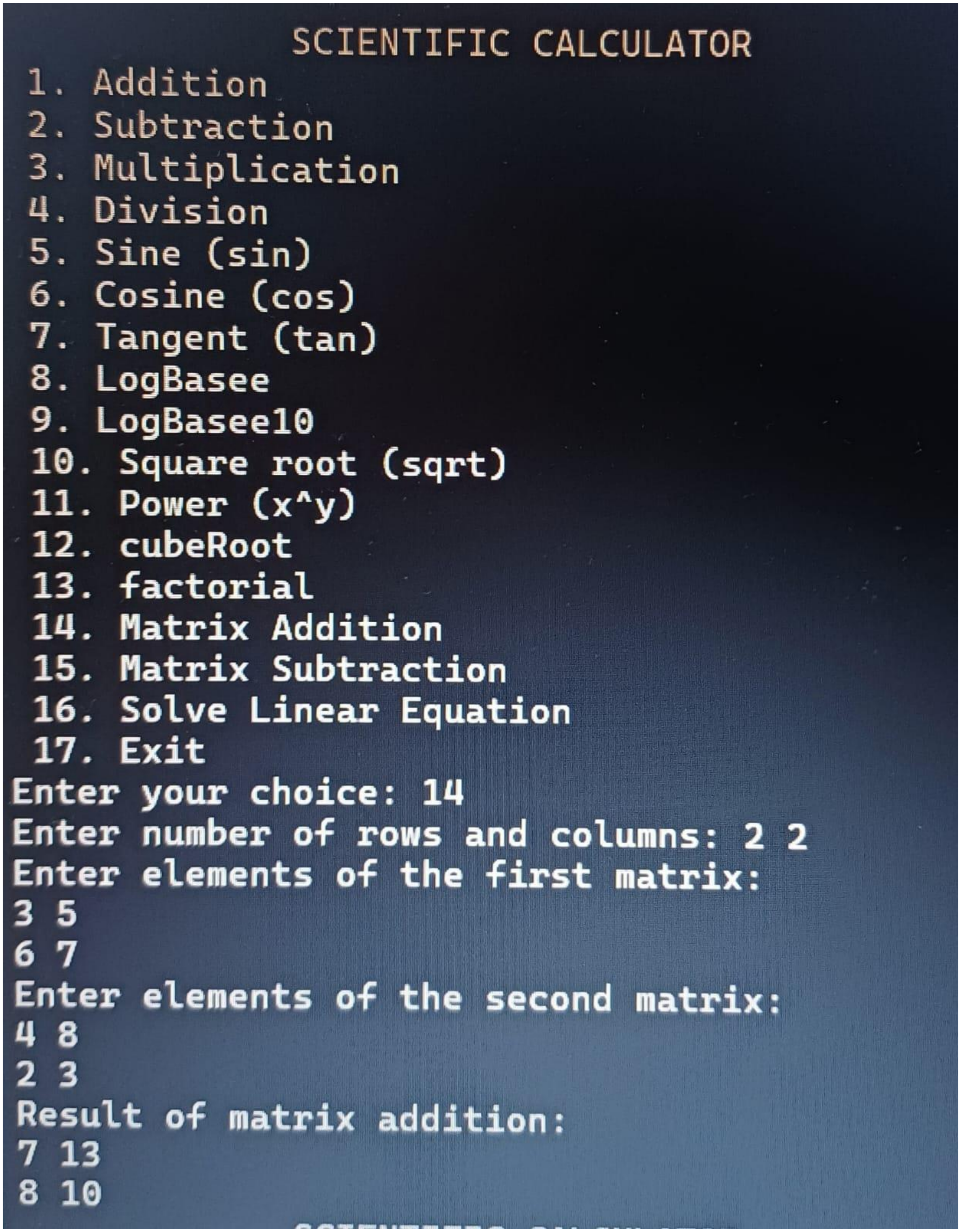
```
              SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 12
Enter a number: 8
Result = 2
```

13.factorial:



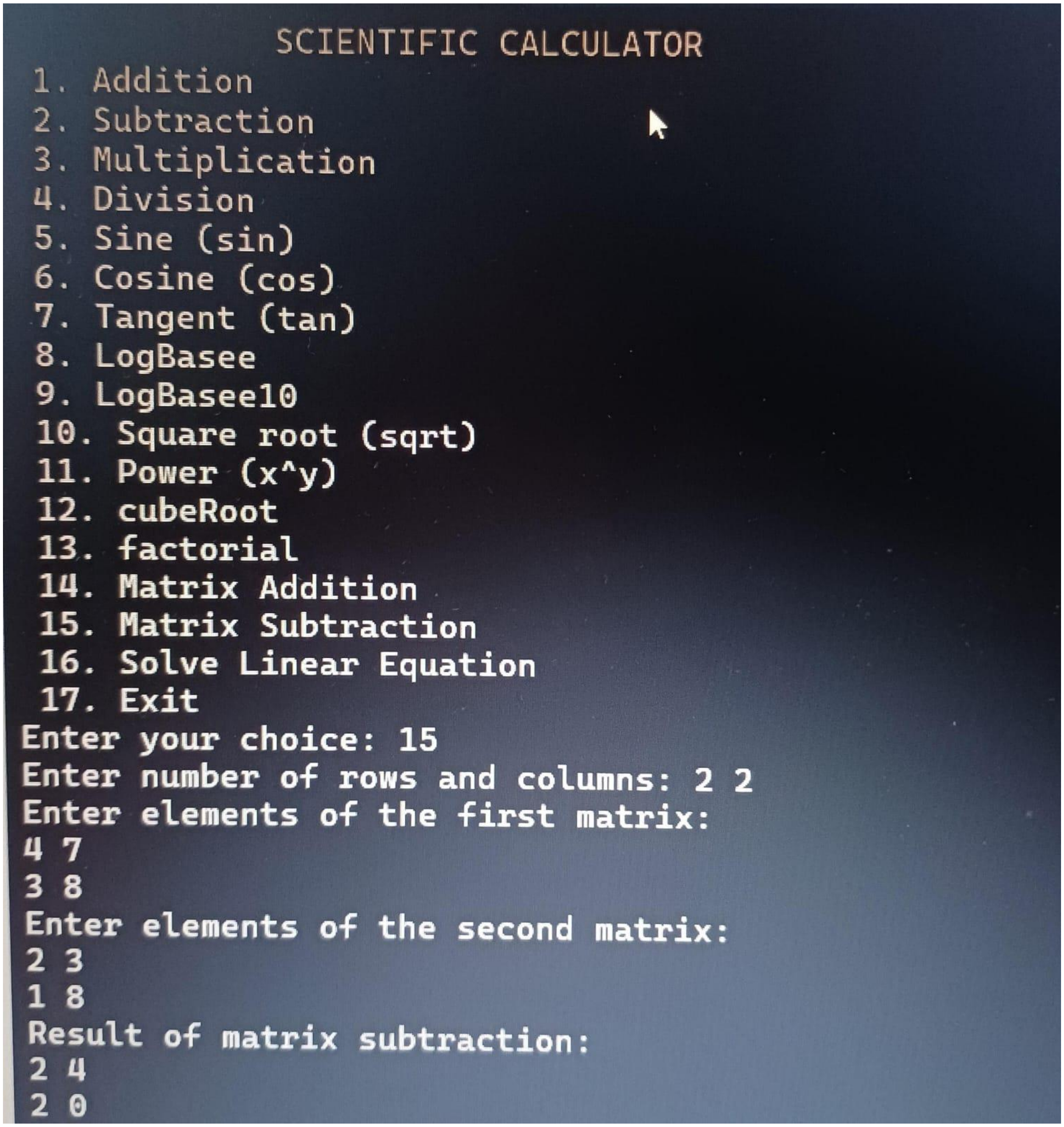
```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 13
Enter an integer: 6
Factorial of 6 = 720
```

14.Matrix addition:



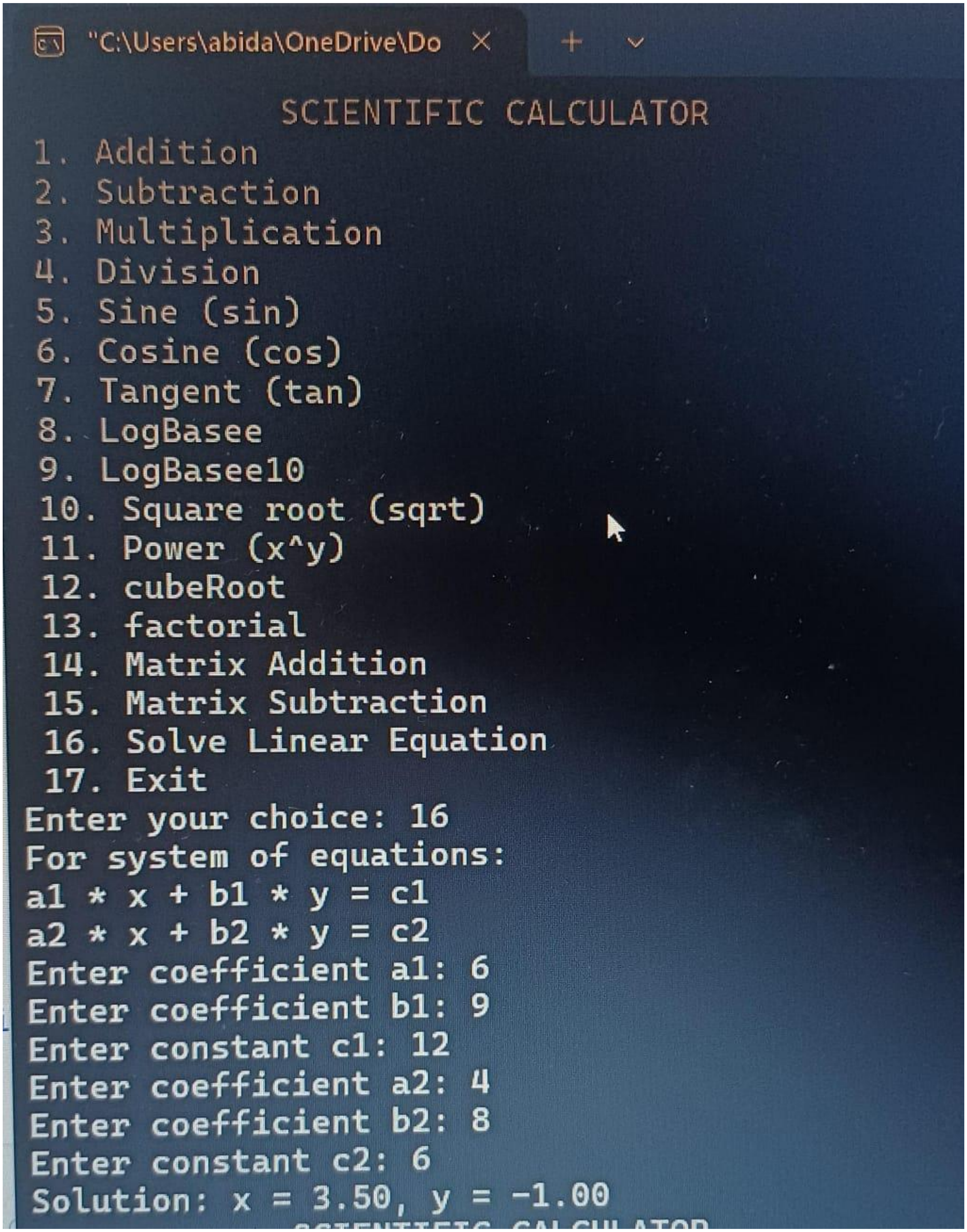
```
             SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 14
Enter number of rows and columns: 2 2
Enter elements of the first matrix:
3 5
6 7
Enter elements of the second matrix:
4 8
2 3
Result of matrix addition:
7 13
8 10
```

15.Matrix substraction:



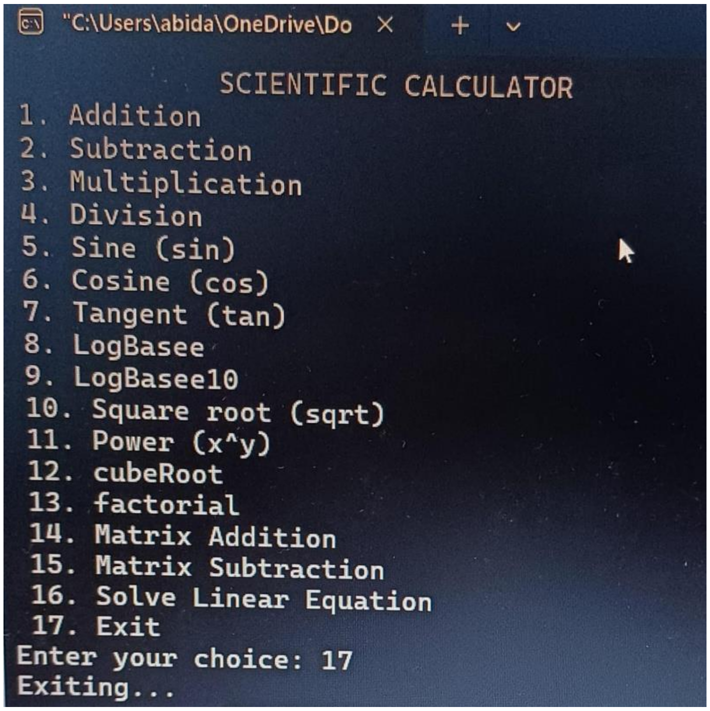
```
              SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 15
Enter number of rows and columns: 2 2
Enter elements of the first matrix:
4 7
3 8
Enter elements of the second matrix:
2 3
1 8
Result of matrix subtraction:
2 4
2 0
```

16.Solve linear equation:



```
                SCIENTIFIC CALCULATOR
 1. Addition
 2. Subtraction
 3. Multiplication
 4. Division
 5. Sine (sin)
 6. Cosine (cos)
 7. Tangent (tan)
 8. LogBasee
 9. LogBasee10
 10. Square root (sqrt)
 11. Power (x^y)
 12. cubeRoot
 13. factorial
 14. Matrix Addition
 15. Matrix Subtraction
 16. Solve Linear Equation
 17. Exit
Enter your choice: 16
For system of equations:
a1 * x + b1 * y = c1
a2 * x + b2 * y = c2
Enter coefficient a1: 6
Enter coefficient b1: 9
Enter constant c1: 12
Enter coefficient a2: 4
Enter coefficient b2: 8
Enter constant c2: 6
Solution: x = 3.50, y = -1.00
```

17.Exit:



```
"C:\Users\abida\OneDrive\Do

              SCIENTIFIC CALCULATOR
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Sine (sin)
6. Cosine (cos)
7. Tangent (tan)
8. LogBasee
9. LogBasee10
10. Square root (sqrt)
11. Power (x^y)
12. cubeRoot
13. factorial
14. Matrix Addition
15. Matrix Subtraction
16. Solve Linear Equation
17. Exit
Enter your choice: 17
Exiting...
```

Limitations of this project:

No Input Validation:

- The program does not validate the input. If the user enters something that is not an integer (e.g., a string or a floating-point number), `scanf()` may fail, and the program may behave unpredictably. For example, if the user inputs a non-integer value, the program may not enter valid data into the array, causing undefined behavior.

Integer Overflow:

- The variable `result` is of type `int`, which means it can only hold values within the range of an `int` (typically −`2,456,356,657`). If the sum of the numbers exceeds this range, it will cause an **integer overflow** and produce incorrect results.

- For example, if the sum exceeds `INT_MAX`, the result will wrap around and could show an incorrect negative number or behave unpredictably.

Memory Efficiency:

- The program stores all the numbers in the array, but in most cases, we only need to keep track of the running total (`result`). Storing all the numbers consumes unnecessary memory. Instead, the program could accumulate the sum directly as the numbers are entered, without storing them in an array.

No Dynamic Memory Management:

The `numbers[]` array is created on the stack, which means it has a limited size. For very large arrays, this can be problematic. The program does not use dynamic memory allocation (e.g., `malloc()`), which would allow the array to be allocated on the heap, offering more flexibility and avoiding stack overflow.

Potential for Undefined Behavior with Invalid `scanf()`:

- The program doesn't check if `scanf()` successfully reads the user input. If `scanf()` fails to read a valid integer (for example, if the user enters a non-numeric value), the array will not be populated correctly, and the program may produce incorrect results or crash.

Future Scope of the Scientific Calculator Project

1.Enhaced Graphing and Visualization:

Add support for 2D and 3D graphing to visualize mathematical functions and solve graph-based problems.

2. Complex Number and Matrix Support:

Integrate support for complex number arithmetic and matrix operations, expanding use cases in engineering and physics.

3. Symbolic Computation and Equation Solving:

Incorporate symbolic computation to handle algebraic expressions and solve equations symbolically (e.g., using libraries like SymPy in Python).

4. Programming Mode:

Add functions like binary, octal, and hexadecimal calculations, bitwise operations, and logical operations useful for programmers.

5. Error Analysis and Accuracy Control:

Add features to handle rounding errors, precision settings, and error propagation analysis, especially for high-precision calculations.

6. Mobile or Web Version:

Expand the project to create mobile or web-based versions to make it accessible across multiple devices.

7. Voice Commands and AI Integration:

Integrate voice recognition for hands-free operation or use AI to interpret more complex queries (e.g., solving word problems).

8. Scientific Data Import and Export:

Allow users to import or export data files (e.g., CSVs) for more complex calculations, useful in scientific research.

Conclusion:

The Scientific Calculator project aims to revolutionize the way students and professionals perform calculations, visualize data, and solve problems. With its blend of advanced mathematical functions, scientific capabilities, and user-friendly design, it promises to be a powerful tool for anyone who needs to perform scientific or engineering calculations quickly and accurately. The project’s focus on versatility, ease of use, and accessibility ensures that it will be a valuable asset for users across many different fields of study and work.