

# CS201 – Spring 2022-2023 - Sabancı University

## Homework #4: Matrix Calculator

Due May 3, Wednesday, 23:00 (Sharp Deadline)

### Introduction

The aim of this homework is to practice on the use of vectors, loops and if-else statements. The use of vectors and loops are necessary in this problem.

Your homework will be automatically graded using SUCourse, so it is very important to satisfy the exact same outputs given in the example test cases of SUCourse. Please submit your assignment by writing your main source (cpp) file content into the Answer field. You can utilize the **Check** button under the code editor at SUCourse to check whether your implementation is working in the expected way. After you check your solution code, you will see your grade with the example test cases used; however your homework will then be graded with **different** test cases.

To submit your homework, you must hit the “**Finish attempt...**” and “**Submit all and finish**” buttons. Just a reminder of a character  which refers to a newline in your expected output.

### Description

In this homework, you will write a C++ program that will provide operations performed by a matrix calculator. There are four different operations that are provided by the calculator. The first operation is the **addition** of two matrices. The second operation is the **multiplication** of two matrices. The third operation is taking the **power** of a single matrix. The final operation is obtaining the **transpose** of a single matrix.

### VERY IMPORTANT!

Your programs will be compiled, executed and evaluated automatically; therefore, you should definitely follow the rules for prompts, inputs and outputs. See Sample Runs section for some examples.

- Order of inputs and outputs must be in the abovementioned format.
- Prompts before inputs and outputs must be exactly the same with examples.

Following these rules is crucial for grading, otherwise our software will not be able to process your outputs and you will lose some grades in the best scenario.

## Input Checks and Program Flow

Firstly, your program will display a menu of operations for the user to choose from. The menu provided will look like this:

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5- Exit

The user should enter a choice from the menu provided. You may assume that the user will enter an integer between the correct range. You do not need to perform any checks on the input entered by the user when choosing an option from the menu.

**\*\*\*Your program should keep providing the menu until the user chooses the Exit option\*\*\***

If the user picks the **addition** operation, the user will be asked to enter the dimensions of the first and second matrices respectively. If the two matrices are compatible for addition, the user will be asked to enter the numbers for the first matrix (row by row) and then display it. The user will then be asked to enter the numbers for the second matrix (row by row) and then display it. Once both matrices are entered, your program should calculate their sum and display the result .

If the user picks the **multiplication** operation, the user will be asked to enter the dimensions of the first and second matrices respectively. If the two matrices are compatible for multiplication, the user will be asked to enter the numbers for the first matrix (row by row) and then display it. The user will then be asked to enter the numbers for the second matrix (row by row) and then display it. Once both matrices are entered, your program should calculate their multiplication and display the result.

If the user picks the **power** operation, the user will be asked to enter the dimensions of a single matrix. If the matrix is compatible for the power operation, the user will be asked to enter the numbers for the single matrix (row by row). The user will then be asked to enter the power that the matrix will be raised to. Once the matrix and the power are entered, your program should perform the calculation and display the result.

**\*\*\*Note that the power (exponent) n given by the user MUST an INTEGER greater than or equal to 1 (n >= 1)\*\*\***

If the user picks the **transpose** operation, the user will be asked to enter the dimensions of a single matrix. The user will then be asked to enter the numbers for the single matrix (row by row). Once the matrix is entered, your program should calculate the transpose matrix and then display it.

When the user is required to enter the dimensions of a matrix, the user is expected to enter two integers. The first integer represents the number of rows of the matrix and the second integer represents the number of columns of the matrix.

**\*\*\*IMPORTANT NOTE: ALL numbers entered to fill the matrices SHOULD be integers\*\*\***

## IMPORTANT!

If your code does not compile, you will get **zero**. Please be careful about this and double check your code before submission.

## Operations

### 1- Addition

To add two matrices, just add the corresponding entries, and place this sum in the corresponding position in the matrix which results.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} a + w & b + x \\ c + y & d + z \end{bmatrix}$$

The following is an example of the addition operation:

$$\begin{bmatrix} 1 & 5 \\ -4 & 3 \end{bmatrix} + \begin{bmatrix} 2 & -1 \\ 4 & -1 \end{bmatrix}$$

The matrices have the same dimensions (2 x 2) so we can add them.

$$\begin{bmatrix} 1 & 5 \\ -4 & 3 \end{bmatrix} + \begin{bmatrix} 2 & -1 \\ 4 & -1 \end{bmatrix} = \begin{bmatrix} 1 + 2 & 5 + (-1) \\ -4 + 4 & 3 + (-1) \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 0 & 2 \end{bmatrix}$$

### 2- Multiplication

To check if the matrices are compatible for multiplication, the number of columns in the first matrix must be equal to the number of rows in the second matrix.

For example, Matrix A (3 x 2) and Matrix B (2 x 3) are compatible for multiplication.

For example, Matrix A (3 x 3) and Matrix B (2 x 3) are not compatible for multiplication.

If the matrices are compatible for multiplication then the resulting matrix will have the same number of rows as the first matrix and the same number of columns as the second matrix.

For example, Matrix A (3 x 2) and Matrix B (2 x 4), the resulting multiplication will yield a matrix with the following dimensions (3 x 4).

To perform the multiplication, you take each row of the first matrix and multiply each column of the second matrix going by each element.

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} w & x \\ y & z \end{bmatrix} = \begin{bmatrix} aw + by & ax + bz \\ cw + dy & cx + dz \end{bmatrix}$$

### 3- Power

To check if the matrix is compatible for the power operation, The number of rows should be equal to the number of columns for that matrix.

For example, Matrix A (2 x 2) is compatible for the power operation.

For example, Matrix B (3 x 2) is not compatible for the power operation.

If n is the exponent, to find the power of a matrix , multiply the matrix by itself n -1 many times.

For example, if the exponent is 2 and we have a matrix:

$$A = \begin{bmatrix} 1 & -3 \\ 2 & 5 \end{bmatrix}$$

$$A^2 = A * A$$

$$A^2 = \begin{bmatrix} 1 & -3 \\ 2 & 5 \end{bmatrix} * \begin{bmatrix} 1 & -3 \\ 2 & 5 \end{bmatrix} = \begin{bmatrix} -5 & -18 \\ 12 & 19 \end{bmatrix}$$

### 4- Transpose

The transpose of a matrix is found by interchanging its rows into columns or columns into rows.

$$A = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}_{(2 \times 3)} \quad A^T = \begin{bmatrix} a & d \\ b & e \\ c & f \end{bmatrix}_{(3 \times 2)}$$

The resulting transpose matrix will have the same number of columns as rows and the same number of rows as columns.

For example, Matrix A (3 x 1) has a Transpose Matrix  $A^T$  (1 x 3)

For example, Matrix B (4 x 2) has a Transpose Matrix  $B^T$  (2 x 4)

The following is an example of the transpose operation:

$$B = \begin{bmatrix} 2 & -9 & 3 \\ 13 & 11 & -17 \\ 3 & 6 & 15 \\ 4 & 13 & 1 \end{bmatrix}_{(4 \times 3)}$$

$$B^T = \begin{bmatrix} 2 & 13 & 3 & 4 \\ -9 & 11 & 6 & 13 \\ 3 & -17 & 15 & 1 \end{bmatrix}_{(3 \times 4)}$$

## Operation Checks

A total of four operations will be provided by your matrix calculator. There are checks that have to be done to check if an operation can be performed. Checks for three operations are provided in Table 1 with their corresponding error messages.

<u>Operation</u>	<u>Check</u>	<u>Message</u>
Addition	Both matrices should have the same dimensions.	The two matrices do not have the same dimensions. Addition cannot be performed.
Multiplication	The number of columns in the first matrix must be equal to the number of rows in the second matrix.	The number of columns in the first matrix must be equal to the number of rows in the second matrix. Multiplication cannot be performed.
Power	The number of rows should be equal to the number of columns for the single matrix.	The matrix does not have the same number of rows as columns. Power cannot be performed.

**Table 1**

## No abrupt program termination please!

You may want to stop the execution of the program at a specific place (before the end) in the program. Although there are ways of doing this in C++, it is not a good programming practice to abruptly stop the execution in the middle of the program. Therefore, your program flow should continue until the end of the main function and finish there.

## Sample Runs

Below, we provide some sample runs of the program that you will develop. The italic and bold phrases are inputs taken from the user. You should follow the input order in these examples and the prompts your program will display must be **exactly the same** as in the following examples.

### Sample Run 1

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5 - Exit

Choice: **1**

Please enter the dimensions of the first matrix: **2 2**

Please enter the dimensions of the second matrix: **3 2**

The two matrices do not have the same dimensions. Addition cannot be performed.

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5 - Exit

Choice: **1**

Please enter the dimensions of the first matrix: **2 2**

Please enter the dimensions of the second matrix: **2 2**

The two matrices have the same dimensions. Addition can be performed.

Please enter the numbers for the first matrix.

Enter the numbers for row #1: **1 2**

Enter the numbers for row #2: **3 4**

The first matrix is:

1 2

3 4

Please enter the numbers for the second matrix.

Enter the numbers for row #1: **1 2**

Enter the numbers for row #2: **10 5**

The second matrix is:

1 2

10 5

Result of the Sum Operation:

2 4

13 9

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5 - Exit

Choice: **5**

## **Sample Run 2**

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5 - Exit

Choice: **2**

Please enter the dimensions of the first matrix: **2 2**

Please enter the dimensions of the second matrix: **5 2**

The number of columns in the first matrix must be equal to the number of rows in the second matrix. Multiplication cannot be performed.

Pick a choice from the list:

- 1- Addition Operation
- 2- Multiplication Operation
- 3- Power Operation
- 4- Transpose Operation
- 5 - Exit

Choice: **2**

Please enter the dimensions of the first matrix: **3 2**

Please enter the dimensions of the second matrix: **2 3**

The number of columns in the first matrix is equal to the number of rows in the second matrix. Multiplication can be performed.

Please enter the numbers for the first matrix.

Enter the numbers for row #1: **10 20**

Enter the numbers for row #2: **1 10**

Enter the numbers for row #3: **3 5**

The first matrix is:

10 20

1 10

3 5

Please enter the numbers for the second matrix.

Enter the numbers for row #1: **10 20 30**

Enter the numbers for row #2: **40 50 60**

The second matrix is:

10 20 30

40 50 60

Result of the Multiplication Operation:

900 1200 1500

410 520 630

230 310 390

Pick a choice from the list:

1- Addition Operation

2- Multiplication Operation

3- Power Operation

4- Transpose Operation

5 - Exit

Choice: **5**

### **Sample Run 3**

Pick a choice from the list:

1- Addition Operation

2- Multiplication Operation

3- Power Operation

4- Transpose Operation

5 - Exit

Choice: **3**

Please enter the dimensions of the matrix: **3 2**

The matrix does not have the same number of rows as columns.

Power cannot be performed.

Pick a choice from the list:

1- Addition Operation

2- Multiplication Operation



3- Power Operation  
4- Transpose Operation  
5 - Exit  
Choice: 3

Please enter the dimensions of the matrix: **2 2**  
The matrix has the same number of rows as columns. Power can be performed.

Please enter the numbers for the matrix.  
Enter the numbers for row #1: **2 2**  
Enter the numbers for row #2: **2 2**  
The matrix is:  
2 2  
2 2

Please enter the power: **3**  
Result of the Power Operation:  
32 32  
32 32

Pick a choice from the list:  
1- Addition Operation  
2- Multiplication Operation  
3- Power Operationcccccc  
4- Transpose Operation  
5 - Exit  
Choice: **5**

### **Sample Run 4**

Pick a choice from the list:  
1- Addition Operation  
2- Multiplication Operation  
3- Power Operation  
4- Transpose Operation  
5 - Exit  
Choice: **4**

Please enter the dimensions of the matrix: **3 3**  
Please enter the numbers for the matrix.  
Enter the numbers for row #1: **1 2 3**

Enter the numbers for row #2: **4 5 6**

Enter the numbers for row #3: **7 8 9**

The matrix is:

1 2 3

4 5 6

7 8 9

Result of the Transpose Operation:

1 4 7

2 5 8

3 6 9

Pick a choice from the list:

1- Addition Operation

2- Multiplication Operation

3- Power Operation

4- Transpose Operation

5 - Exit

Choice: **5**

## **General Rules and Guidelines about Homework**

The following rules and guidelines will be applicable to all homework unless otherwise noted.

### **How to get help?**

You may ask questions to TAs (Teaching Assistants) or LAs (Learning Assistants) of CS201. Office hours of TAs/LAs are at the SUCourse.

### **What and Where to Submit**

You can prepare (or at least test) your program using MS Visual Studio C++ (Windows users) or using XCode (macOS users).

- Your code will be automatically graded using SUCourse. Therefore, it is essential that you ensure your output matches the exact same outputs given in the example test cases provided by SUCourse.
- After writing your code, use the "Check" button located under the code editor in SUCourse to see your grade based on the example test cases used. This grade will give you an idea of how well your code is performing.
- Note that the example test cases used for checking your code are not the same as the ones used for grading your homework. Your final grade will be based on different test cases. Therefore, it is important that you carefully follow the instructions and ensure that your code is working correctly to achieve the best possible grade on your homework assignment.
- To submit your homework, click on the "Finish attempt..." button and then the "Submit all and finish" button. If you wish to submit again before the due date, you can press the "Re-attempt quiz" button.

- Submit your work **through SUCourse only!** You will receive no credits if you submit by any other means (email, paper, etc.).

### **Grading, Review and Objections**

Be careful about the automatic grading: Your programs will be graded using an automated system. Therefore, you should follow the guidelines on the input and output order. Moreover, It is important to use the exact same text as provided in the example test case outputs from SUCourse. Otherwise, the automated grading process will fail for your homework, and you may get a zero, or in the best scenario, you will lose points.

#### Grading:

- There is NO late submission. You need to submit your homework before the deadline. Please be careful that SUCourse time and your computer time may have 1-2 minute differences. You need to take this time difference into consideration.
- Successful submission is one of the requirements of the homework. If, for some reason, you cannot successfully submit your homework and we cannot grade it, your grade will be 0.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work **only**. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the [Syllabus](#).

## **Plagiarism will not be tolerated!**

Grade announcements: Grades will be posted in SUCourse, and you will get an Announcement at the same time. You will find the grading policy and test cases in that announcement.

Grade objections: It is your right to object to your grade if you think there is a problem, but before making an objection please try the steps below and if you still think there is a problem, contact the TA that graded your homework from the email address provided in the comment section of your announced homework grade or attend the specified objection hour in your grade announcement.

- Check the comment section in the homework tab to see the problem with your homework.
- Check the test cases in the announcement and try them with your code.
- Compare your results with the given results in the announcement.

***Good Luck!***

***Mohamed Zeina, Erchan Aptoula and Gülşen Demiröz***