Faculty of Engineering
Computer and Systems Engineering Dept.

First Year Electrical Engineering
CSE121- Computer Programming
2018/2019 Second Semester

Project

Due Date for Project Submission:

Level1 : Saturday May 11th, 2019 (till 11:59PM).

All remaining levels: Friday June 28th , 2019 (till 11:59PM)

Teams: No Groups, Work is individual based (Each Student submit a project).

Notes:

- Automatic Grading will be applied over an online system at (**cseasu.com**).
- Use the Submission Manual that explains how to register on the online system and submit, test/evaluate your code.
- The online system can automatically detect copied submissions with intelligent comparison (plagiarism detection) even with previous years projects. All detected copies of students submissions will take **negative** grade.

Project Description

It is required to develop a program to do Matrix operations. The program use a defined string format to represent the matrix in the user input and output sections. For the following matrix:

$$\begin{bmatrix} 10 & 2.13 & 3 \\ -5 & 0 & 4 \\ 16.5 & 1 & 8 \end{bmatrix}$$

the string representation of the Matrix is :

[10 2.13 3;-5 0 4;16.5 1 8]

In the program, the user enters a matrix in the defined string format then asked to enter an operator from the following list:

Level	operator	Meaning	Action Required from the Program
1	+	Plus to another Matrix	Reads from the user another matrix in the defined format, do the **ADD** operation and print the result in the same matrix format.
	-	Minus to another Matrix	Reads from the user another matrix in the defined format, do the **SUB** operation and print the result in the same matrix format.
	*	Multiply to another Matrix	Reads from the user another matrix in the defined format, do the **Multiply** operation and print the result in the same matrix format.
2	^	Power	Reads from the user a positive integer number (x). Calculate the matrix to the **power** (x) and print

			the result in the same matrix format.
	T	Transpose	No extra input is needed from the User. Calculate the **transpose** of the matrix and print the result in the same matrix format.
	D	Determinant	No extra input is needed from the User. Calculate the **Determinant** value of the matrix and print the result value.
3	I	Inverse	No extra input is needed from the User. Calculate the **inverse** of the matrix and print the result in the same matrix format.
	/	Divide by another Matrix	Reads from the user another matrix in the defined format, do the **Divide** operation as Multiply between the first matrix and inverse of the second matrix and print the result in the same matrix format.

Extra Marks:

Support complex values in the matrix elements.

Project Marks Distribution:

60% to support operators in Level 1

20% to support operators in Level 2

20% to support operators in Level 3

20% for Extra Marks Part

General Constraints:

1- User can input matrices of any size up to 100 x 100.
2- All values are of float data types.
3- Do not prompt user to enter anything , just read the input directly.
4- Any error in user input, the program should output the word “ERROR” in uppercase letters and exits. List of Errors includes the following:
    a. Any missing or unexpected input in the entered string matrix.
    b. Invalid corresponding matrices size in +, -, * , / operators.
    c. Non square matrix in Inverse and Power operators.
    d. Non integer positive power in power operator.
5- Output should not include any extra white spaces or any extra texts more than the results.

Hints:

1. Define 2 dimensional arrays of size 100 x 100 and use only the required size in the operation.

2. While parsing the matrix, you will need to convert from string to float and from float to string. Use the following example as a guide:

```
#include<iostream>
#include<string>
#include<cstdlib>
#include<sstream>

using namespace std;

int main() {
// To convert from string to float use atof function
// atof needs #include<cstdlib>

string s = "1.5";
float v;
v = atof(s.c_str());


// To convert float-to string use ostringstream variable
// ostringstream needs #include<sstream>
// ostringstream variables are used in a similar way to cout
// Then get the string from the ostringstream using .str

float d = 1.55;
string str;
ostringstream ss;
ss << d;
str = ss.str();


return 0;
}
```

3. To support complex numbers (in the Extra Marks section), you might represent each matrix with 2 arrays each of them is of 2 dimensional

type. The first to hold the real part of the number and the second to hold the imaginary part of the number.

Input / Output Samples

User Input in Blue, Program Output in Red (Just for Clarification, not required in the program)

Example 1 (Add Operator)

```
[1 -2.5 3;4 5.25 6;7 8 9.12]
+
[3 4.2 10.2;12 -1 0;67 2 13]
[4 1.7 13.2;16 4.25 6;74 10 22.12]
```

Example 2 (Power Operator)

```
[1 -2.5;4 5.2]
^
2
[-9 -15.5;24.8 17.04]
```

Example 3 (Transpose Operator)

```
[1 -2.5;4 5.2]
T
[1 4;-2.5 5.2]
```

Example 4 (Determinant Operator)

```
[1 -2.5;4 5.2]
D
15.2
```

Example 5 (Inverse Operator)

```
[1 -2.5;4 5.2]
I
[0.3421 0.1645;-0.2632 0.0658]
```

Example 6 (error in the input matrix format, inconsistent number of columns)

```
[1 -2.5 10;4 5.2]
ERROR
```

Example 7 (error in the operator requirement, non-square matrix is invalid for Determinant operator)

```
[1 -2.5 10;4 5.2 12]
D
ERROR
```

Example 8 (error in the operator requirement, number of columns of the first matrix is not the same as number of rows of the second matrix, this is invalid for Multiply operator)

```
[1 -2.5 3;4 5.25 6;7 8 9.12]
*
[3 4.2 10.2;12 -1 0]
ERROR
```