Basics: From C to C++

Computer Programming for Engineers (DSAF003-42)
Fall, 2021

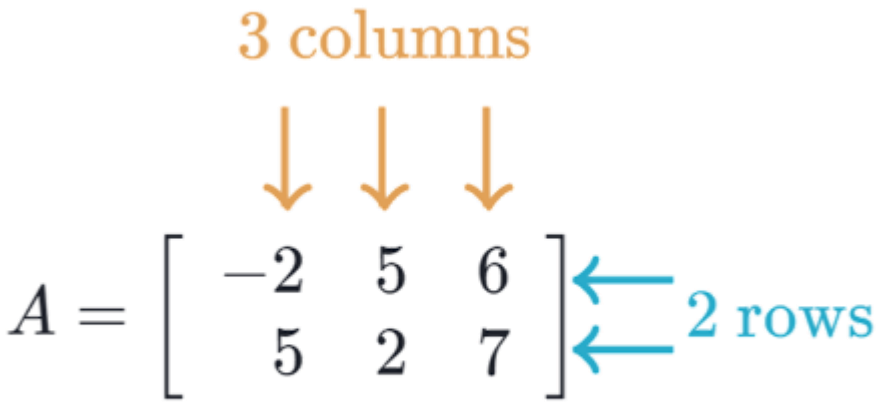
Assignment : PA3

Instructor:
Youngjoong Ko (nlp.skku.edu)

PA3

- **In mathematics, a Matrix is a rectangular array or table of numbers, symbols, or expressions, arranged in rows and columns, which is used to represent a mathematical object or a property of such an object. (Wikipedia)**

- **For examples,**



A is 2 X 3 Matrix

PA3

- **The goal of PA3 is implementation Matrix class and three Matrix Operations.**

- **Matrix Operations**
  1. Addition
  2. Subtraction
  3. Multiplication

Matrix Operatioin

- **Addition**

The addition A + B of two M x N Matrices A and B is calculated entrywise:

$$(A+B)_{i,j} = A_{i,j} + B_{i,j} \; where \; 1 \le i \le M \; and \; 1 \le j \le N.$$

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{4} \end{bmatrix} + \begin{bmatrix} \mathbf{5} & \mathbf{6} \\ \mathbf{7} & \mathbf{8} \end{bmatrix} = \begin{bmatrix} \mathbf{1}+\mathbf{5} & \mathbf{2}+\mathbf{6} \\ \mathbf{3}+\mathbf{7} & \mathbf{4}+\mathbf{8} \end{bmatrix} = \begin{bmatrix} \mathbf{6} & \mathbf{8} \\ \mathbf{10} & \mathbf{12} \end{bmatrix}$$

Matrix Operatioin

- **Subtraction**

The subtraction A - B of two M x N Matrices A and B is calculated entrywise:

$$(A - B)_{i,j} = A_{i,j} - B_{i,j} \; where \; 1 \leq i \leq M \; and \; 1 \leq j \leq N.$$

$$\begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{4} \end{bmatrix} - \begin{bmatrix} \mathbf{4} & \mathbf{6} \\ \mathbf{8} & \mathbf{10} \end{bmatrix} = \begin{bmatrix} \mathbf{1-4} & \mathbf{2-6} \\ \mathbf{3-8} & \mathbf{4-10} \end{bmatrix} = \begin{bmatrix} \mathbf{-3} & \mathbf{-4} \\ \mathbf{-5} & \mathbf{-6} \end{bmatrix}$$

Matrix Operatioin

- **Multiplication (Scalar)**

The product cA of a scalar c and a Matrix A is computed by multipying every entry of A by c:

$$(cA)_{i,j} = \ c \cdot A_{i,j} \ where \ 1 \leq i \leq M \ and \ 1 \leq j \leq P.$$

$$5 \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 \times 1 & 5 \times 2 \\ 5 \times 3 & 5 \times 4 \end{bmatrix} = \begin{bmatrix} 5 & 10 \\ 15 & 20 \end{bmatrix}$$

Matrix Operatioin

- **Multiplication (Matrix)**

If A is an M x N Matrix and B is an N x P Matrix, the their matrix product AB is the M X P Matrix whose entries are given by dot product of the corresponding row of A and the coressponding column of B:

$$[AB]_{i,j} = \sum_{r=1}^{N} a_{i,r} b_{r,j} \ \ where\ 1 \leq i \leq M\ and\ 1 \leq j \leq P.$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1 \times 5 + 2 \times 7 & 1 \times 6 + 2 \times 8 \\ 3 \times 5 + 4 \times 7 & 3 \times 6 + 4 \times 8 \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Main Functions

- In Main Function, Get Two Operands and one Operator

- If Operand is Matrix, Number of Row and column are given as integers.
- If Operand is Scalar, Scalar value are given as integers.

- There are 4 Operators

Input Command	Operator
+	Addition
-	Subtraction
*	Scalar Multiplication
x	Matrix Multiplication

Matrix Class

- **Variables**
  - Two integer variable for number of rows and columns
  - 2-D integer Array for save Matrix

- **Have to Implement**
  - Constructor
  - print() function for printing Matrix
  - 3 Operators

    \+ : Addition

    \- : Subtraction

    * : Multiplication (Scalar, Matrix both)

Must follow rules

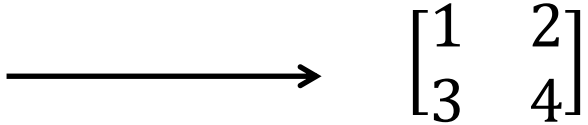
- **Use Same Operator for implement Scalar Multiplication and Matrix Multiplication.**

- **If shape doesn't match, Print Error Message.**
  - Addition and Subtraction should match each row and column
  - Matrix multiplication should match left matrix's column and right matrix's row

- **The N x M Matrix will be given as a sequence with N x M numbers divided by " ".**

  e.g. 2 x 2 Matrix

  Sequence : 1 2 3 4

- **Use cout format in slide 15.**

Examples

- **Addition**
  - If two Matrices have same shape

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : +
Matrix Shape : 2 2
Matrix : 5 6 7 8
Matrix Operation Results
6 8
10 12
```

  - If two Matrices have diffirent shape

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : +
Matrix Shape : 2 3
Matrix : 2 3 4 5 6 7
Not matched shape : (2,2) + (2,3)
```

Examples

- **Subtraction**
  - If two Matrices have same shape

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : -
Matrix Shape : 2 2
Matrix : 0 1 2 3
Matrix Operation Results
1 1
1 1
```

  - If two Matrices have diffirent shape

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : -
Matrix Shape : 2 3
Matrix : 0 1 2 3 4 5
Not matched shape : (2,2) - (2,3)
```

Examples

- **Multiplication (Matrix)**
  - If number of 1st Matrix's columns is equal to number of 2nd Matrix's rows

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : *
Matrix Shape : 2 3
Matrix : 5 6 7 8 9 10
Matrix Operation Results
21 24 27
47 54 61
```

  - If number of 1st Matrix's columns is inequal to number of 2nd Matrix's rows

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : *
Matrix Shape : 3 2
Matrix : 0 1 2 3 4 5
Not matched shape : (2,2) * (3,2)
```

Examples

- **Multiplication (Scalar)**

```
> Matrix Shape : 2 2
Matrix : 1 2 3 4
Operation : x
Int : 5
Matrix Operation Results
5 10
15 20
```

cout format

- **Use follow formats to print. (If you have to print some variables value, fill empty space)**

```
cout << "Matrix Shape : ";
cout << "Matrix : ";
cout << "Operation : ";
cout << "Int : ";
cout << "Matrix Operation Results" << endl;
```

< Error Message >

```
cout << "Not matched shape : (" <<  << "," <<  << ")" << " + " << "("<<  << "," <<  << ")" << endl;
cout << "Not matched shape : (" <<  << "," <<  << ")" << " - " << "("<<  << "," <<  << ")" << endl;
cout << "Not matched shape : (" <<  << "," <<  << ")" << " * " << "("<<  << "," <<  << ")" << endl;
```