# Exercise H

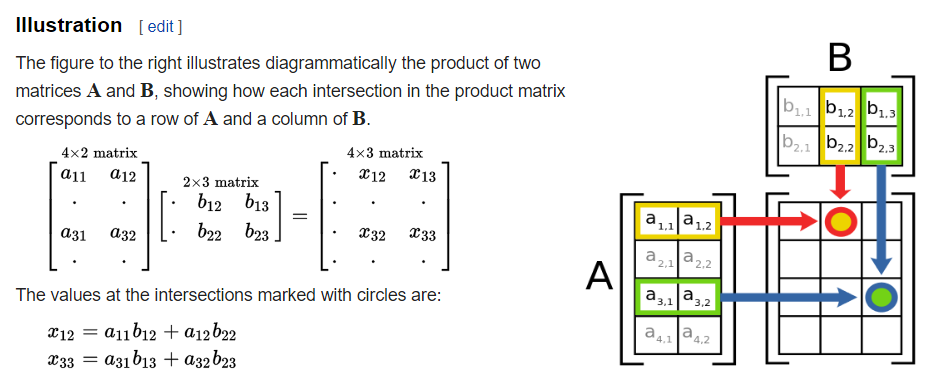
**Section H. Examples involving Modularization using Static Methods**

Note:  *For the following exercises it is possible that an equivalent method is available in the .NET api. However, as an exercise you are required to procedurally code these methods for practice and submission.*

1. Write a static method ReadInteger(string message) that would return an integer. The method should prompt the user with the message, get the input from the user using Console.ReadLine and if the input can be parsed into integer return the integer. If the input cannot be parsed into integer, the method should repeat the prompt until the user enter an integer input.
2. Write a static method PrintArray(int[] arr) that doesn’t return any value. This method should print out all the elements of an array to the console in any format that you have defined in the medthod PrintArray(int[] arr).
3. Write a static method: Substitute(string s, char c1, char c2) that would return a string. The method should find all occurrences of the character c1 in the string s and substitute c1 with character c2. The new word so formed would be the return value of this method.
4. Write a static method: SetArray(int[] arr, int value) that would assign the value into all the elements of the array arr.
5. Write a static method: ResizeArray(int[] arr, int newSize) that would return a new array with the new size and copy all the content of the old array to the new array.
6. Write a static method: IsPrime(int n) that would return a boolean and use IsPrime method to print out prime number from 5 to 1000

**Optional questions**

1. Write a static method MatrixMultiply(int[,] A, int [,] B) that will perform a matrix multiplication and return another 2 dimensional array. Matrix multiplication is done as follows:



(Reference: Wikipedia)

1. Declare a delegate: double DoubleOps(double x) that represent method that will perform some double operation.

Write a static method ProcessArray(double[] arr, DoubleOps ops) that will return an array that has the same size as arr, and apply the delegate on each of the element of arr and assign it to the corresponding elements in the new array.

Use ProcessArray method to get a new array that contains the square root of the elements of the original array and a new array that contains the square of the original array.