

Homework2:

Due Date: Homework2 is due by 11:59 PM on Sunday, November 1, 2020. Submit your work (the .java source code files ONLY, not the compiled .class files!) through the “Homework2” link on Blackboard. You may submit an unlimited number of times; we will only grade the last/latest submission attempt, but be sure to attach all of your files to each submission attempt. Be sure to include your name and Stony Brook ID number in a comment at the beginning of each file that you submit.

Instructions: This assignment is worth 40 points (10 points per program).

1. Write a method that checks whether the input string or a sentence (a string with spaces) is a palindrome or not. The method should be case insensitive and should ignore spaces. Write a test program that prompts the user to input a string and invokes this method. Some example runs are:

Enter the input string: madam
Input string madam is a palindrome

Enter the input string: banana
Input string banana is NOT a palindrome

Enter the input string: Race Car
Input string Race Car is a palindrome

Enter the input string: Too HOT to hoot
Input string Too HOT to hoot is a palindrome

2. Two strings are anagrams if they are written using the same exact letters. Write a method to check if given two strings are anagrams or not. You have to ignore the case and space characters. Write a test program for that prompts the user to input two strings and invokes this method. Some example runs are:

Enter the first string: abba
Enter the second string: abba
abba and abba are anagrams

Enter the first string: banana
Enter the second string: cabana
banana and cabana are NOT anagrams

Enter the first string: Eleven plus two
Enter the second string: Twelve plus one
Eleven plus two and Twelve plus one are anagrams

3. Write down a sort method to sort an array of String using bubble-sort algorithm. Bubble sort algorithm makes several passes through the array. On each pass, successive neighboring pairs are compared. If a pair is not in order, its values are swapped; otherwise the value remains unchanged. The technique is called bubble sort or sinking sort because the smaller values gradually “bubble” their way to the top, and the larger values “sink” to the bottom.

Write a test program that reads in 10 String values and invokes the bubble sort method. Display the sorted string array.

Example run:

Enter 10 strings: New York City, Austin, Dallas, Seattle, Washington D.C., Houston, Chicago, Las Vegas, Charlotte, Denver,

Sorted strings: Austin, Charlotte, Chicago, Dallas, Denver, Houston, Las Vegas, New York City, Seattle, Washington D.C.,

4. Write a method to multiply two matrices. The header of the method is:

public static double[][] multiplyMatrix(double [][] a, double [][] b).

To multiply matrix a by matrix b , the number of columns in a must be the same as the number of rows in b , and the two matrices must have elements of the same or compatible types. Let c be the result of the multiplication. Assume the column size of matrix a is n . Each element $c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + \cdots + a_{in} \times b_{nj}$.

For example, the two 3×3 matrices a and b , c is:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

Where $c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + a_{i3} \times b_{3j}$.

Write a test program that prompts the user to enter two 3×3 matrices and displays their product. Here is a sample run:

Enter matrix1: 1 2 3 4 5 6 7 8 9

Enter matrix2: 0 2 4 1 4.5 2.2 1.1 4.3 5.2

Multiplication of the matrices is:

5.3	23.9	24
11.6	56.3	58.2
17.9	88.7	92.4