# Use arrays to structure the raw data and to perform data comparison & operations

Write a program which sorts ‘N’ array of Strings in an ascending order.

Note: Write two methods:

1. First method: sort using bubble sort
2. Second Method: sort using insertion sort

# Use arrays to structure the raw data and to perform data comparison & operations

Write a program that computes your initials from your full name and displays them.

Example: If name = “KaviArasuV”, it must record the initials as “KAV”

# Use arrays to structure the raw data and to perform data comparison & operations

Write a program which generates a password for a student using his initials and age. Each student will have first name, middle name and last name.

# Use arrays to structure the raw data and to perform data comparison & operations

We have 2 student names Ali Al-Ali and Ahmed Al-Ahmed. Design and implement a Java program that will exchange the last name of the two students in such a way that the new names are going to be

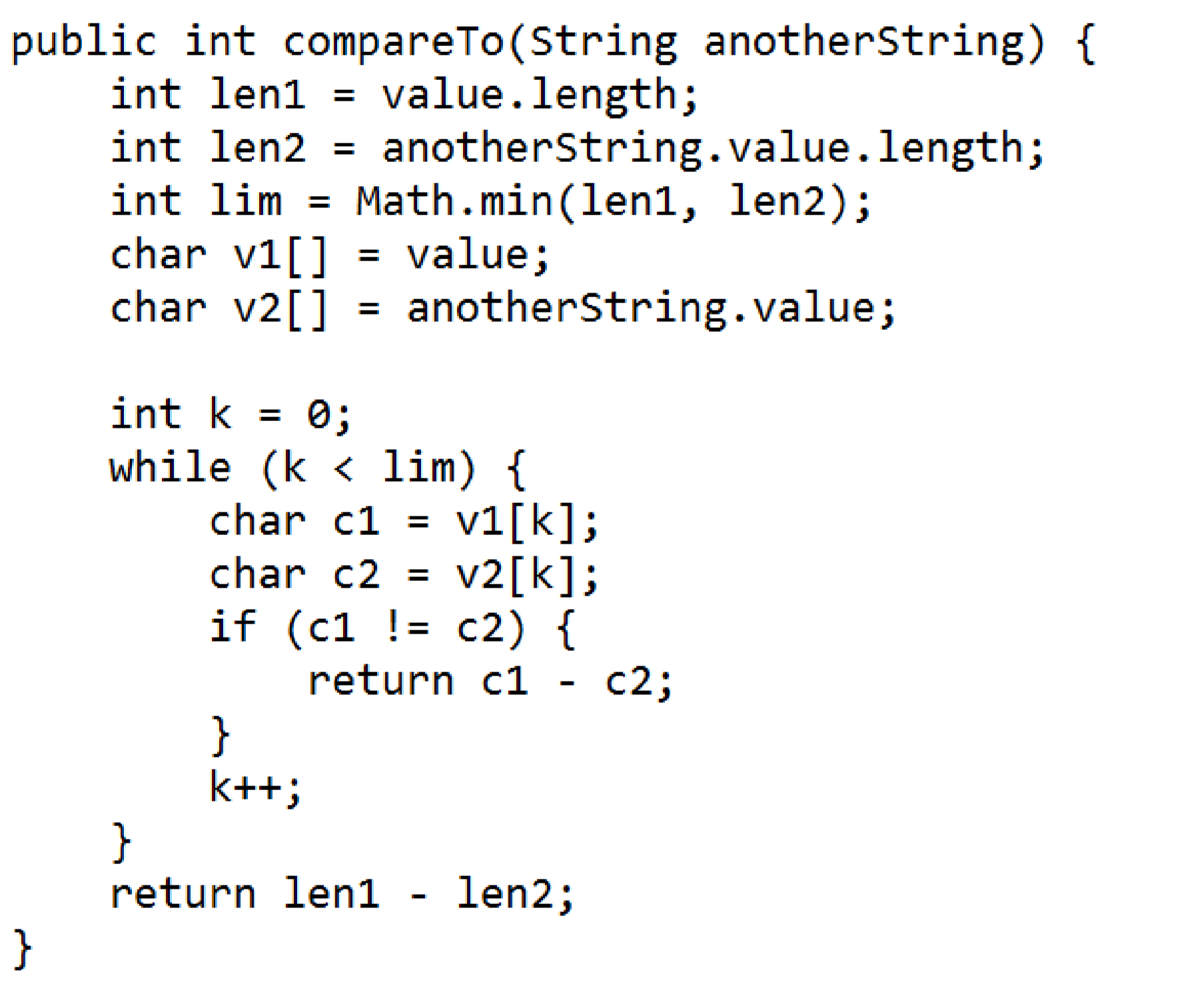
**Ali Al-Ahmed and Ahmed Al-Ali.**

**NOTE:** Your program should work for any names NOT ONLY these given names in the exercise.

# Use arrays to structure the raw data and to perform data comparison & operations

Analyze the below code and invoke the method using main method.

**Use arrays to structure the raw data and to perform data comparison &**



# operations

In an online application, a university would like to validate the university seat number (USN) entered by its student.

Example: A sample USN looks like: 1DS09CS010

Following are the constraints to enter registration number:

* Each USN must be length of 10 characters
* 1st character must be digit and have value either '1' or '2'
* 2nd and 3rd characters must be upper case letters
* 4th and 5th characters must be digits and can have values between 0-9
* 6th and 7th characters must be upper case letters and can have combination any of following substring: CS, IS,EC and ME
* 8th,9th and 10th characters must be digits and can have values between 0-9  Create class called TestUSN which has main() method to:
* Initialize USN variable with string value
* Validate USN against above constraints
* Print "Success" or "Failure" message based on result of validation

# Use arrays to structure the raw data and to perform data comparison & operations

Given a sentence which consists of alphabets [a-zA-Z], digits [0-9], “,” and “.”.

Write a method which returns a string after reversing only the alphabets [a-zA-Z] in a sentence which is passed as an argument the method.

Note: reversing has to be done word-by-word. Example: if sentence contains “Hello World”, the reversed sentence will be “dlroW olleH”

|  |  |  |
| --- | --- | --- |
| **Example** | **Input** | **Output** |
| 1 | 1 cup of hot coffee costs 8.00, whereas cold coffee costs 45.00. | 1 puc fo toh eeffoc stsoc 8.00, saerehw dloc eeffoc stsoc 45.00. |
| 2 | It Costs 25000rs for 1 LCD Projector. | tI stsoC 25000sr rof 1 DCL rotcejorP. |
| 3 | 8990.33 | 8990.33 |

# Use arrays to structure the raw data and to perform data comparison & operations

Complete the method to print the consecutive characters and the number of times appearing in a String.

void printConsecutiveCharacters(String input) {

}

|  |  |  |
| --- | --- | --- |
| **Example** | **Sample Input** | **Sample Output** |
| 1 | “I saw a CD play-er and a modem in ccd” | CD 2 DE 1 |
| 2 | “Student List do not exist in sys-tem” | ST 4  DE 1  NO 1 |

# Use arrays to structure the raw data and to perform data comparison & operations

Let us design a simple compression algorithm where only the frequency of individual letters is used to compress the data. For e.g., the string Aabcccccaaa would become a2b1c5a3. The compression logic should be applied only when the total length of the compressed string is less than the original string. For the purpose of compression logic, the case sensitiveness is not considered. For e.g. A and a are considered the same.

**Input**

A String that needs to be compressed is given. Given string always contains characters. The string may contain characters in upper as well as lower case.

**Output**

Output is the compressed string or the same string if the compressed string length is more than or equal to the length of the original string. Output contains string in lowercase always.

**Input: aAbcccccaaA output: a2b1c5a3**

**Input: BBBBbbb output: b7**

# Use arrays to structure the raw data and to perform data comparison & operations

Triplets are a set of three similar things.

Complete the function to print all the triplets <A, B, C> such that A+B = C

void printTriplets(int[ ] data) {

}

|  |  |  |
| --- | --- | --- |
| **Example** | **Sample Input** | **Sample Output** |
| 1 | data ={2,3,4,5,7} | <2,3,5>  <2,5,7>  <3,4,7> |
| 2 | data = {1,2,3,4,5,7,9} | <1,2,3>  <1,3,4>  <1,4,5>  <2,3,5>  <2,5,7>  <3,4,7>  <4,5,9> |

Use arrays to structure the raw data and to perform data comparison & operations

A retail store wants to keep track of item id and item price of the five items sold by them. Based on the item purchased by the customer, item price must be identified and the computation of bill amount must be done as per the price and quantity of the item purchased.

Write a program to implement the above scenario.

1. Represent the item ids and price in arrays.
2. Search for the item purchased by the customer (assume it to be 5001) in the item ids arrays and identify the respective price. Display and appropriate error message if the item is not found in the array.
3. If the item is found-
4. compute the bill amount as quantity purchased \* price identified.
5. display the bill id, customer id, purchase id, quantity purchased, discount and bill amount.

4- Change the purchase item id value to 5006 and run the program again and observe the result.