|  |  |
| --- | --- |
| **Core Programming** | 2019 |

**Question1 (Find Key)**

**Capabilities covered:**

* Use programming elements to store, load and clear data
* Use arrays to structure the raw data and to perform data comparison & operations

Apply the following rules on a given a list of numbers. The following rules should be applied in the order given below.

***Rules***

a) 1 is added to the middle digit of each number

b) First and the second digits are interchanged

**Input**

A list of n numbers where 3≤n≤9. Each of the given numbers will always be having same number of digits say ‘l’ and l can take values as 3, 5, 7 and 9.

**Output**

Output is the third digit from the beginning of the smallest number after applying the given rules

|  |  |  |
| --- | --- | --- |
| **Test case** | **Input** | **Output** |
| **UTC01\_01** | 386, 752, 961, 573, 839 | 9 |
| **UTC01\_02** | 123456789, 987654321 | 3 |
| **UTC01\_03** | 38765, 21743, 67628 | 8 |

**Assumption & Hint**

Given these numbers 123456789, 987654321. Here n=2 and l=9.

After Rule (a) 1234**6**6789, 9876**6**4321

After Rule (b) **21**346789, **89**7664321

Smallest number is the first one, so the output will be 3 (third digit from the beginning of the smallest number)

**Question 2 (Max Vowels Occurrences)**

**Capabilities covered:**

* Use programming elements to store, load and clear data
* Use arrays to structure the raw data and to perform data comparison & operations

Consider a sentence to find out the word which satisfies the following conditions.

a. Check the word for the presence of all vowels.

b. If two words have all the vowels then, give preference for the string with min length.

c. If more than one word satisfies both the conditions mentioned above, give preference for string which will be first in alphabetical order.

d. If none of the words contains all the vowels then throw a user defined exception named **MissingVowelsException** with message **“Not all vowels”.**

**Input**

A sentence is provided in the form of string.

**Output**

Output is the word from the given sentence, which has all the vowels in it.

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Input** | **Output** |
| **UTC02\_01** | “All the authorities are multimillionare” | “authorities” |
| **UTC02\_02** | “Kindly authorize the technical education ” | “authorize” |
| **UTC2\_03** | “Not all vowels in this statement” | MissingVowelsException with message ”Not all vowels” |

# Question 3 [SQL Query]

**Capabilities covered:**

* Write select queries to retrieve specific set of data from given database

# 

**Lowest price items**

Multiple suppliers offer various prices for the same product. The following ‘product supplier’ table stores the product and its supplier details along with their offer price per unit.



**Description**

1. Write a SQL query that lists all the products along with all those suppliers which offer the lowest offer price. It should also list the unit price.
2. The output should be ordered (ascending) by Product ID
3. There can be more than one supplier who offers the lowest per unit price for a given product. For e.g. SupplierID 1 and 3 have the lowest offer price for productID 1. So, both SupplierID will get listed in the query output which will be ordered by SupplierID

**Assumptions / Hint**

Sample output based on the sample data given in the above table.



**4. String Compression**

**Capabilities covered:**

* Use programming elements to store, load and clear data
* 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