1. Alya, a computer science teacher, gives an assignment to her students. She told students to store the numbers in the array and give a position from the array as input. The students have to replicate the number in that position and display all the numbers along with the replicated number. Help them to do this by using the concept of Arrays.

To do this, create a public class UserInterface with a method getDuplicateElement as follows:

public String getDuplicateElement (): This method should do the following.

Get the size of an array as input, and then get the elements of the array (all elements are int) as input. Next, the user should provide the index of the array. This method should return the element at that index as "**The array elements are <array element> <replicated element>**"

This program may generate an ArrayIndexOutOfBoundsException or InputMismatchException or NegativeArraySizeException.

In the case of an ArrayIndexOutOfBoundsException, the function should return "**Array index is out of range"**.

When providing the input, if the input is not an integer, it will generate an InputMismatchException. In this case, the function should return **"Input was not in the correct format"**.

When providing the input, if the array size is negative, it will generate a NegativeArraySizeException. In this case, the function should return **"Array size should be positive"**.

Use an exception handling mechanism to handle the exception. Use a separate catch block for handling each exception. In the catch block, return the appropriate message.

Write a main method and test the above function.

**Note:**

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Do not use System.exit(0) to terminate the program.**

**Sample Input/Output 1:**

Enter the size of an array

**4**

Enter the array elements

**10**

**20**

**30**

**40**

Enter the position of the element to be replicated

**3**

The array elements are 10 20 30 40 40

**Sample Input/Output 2:**

Enter the size of an array

**3**

Enter the array elements

**2**

**27**

**3**

Enter the position of the element to be replicated

**4**

Array index is out of range

**Sample Input/Output 3:**

Enter the size of an array

**2**

Enter the array elements

**2**

**p**

Input was not in the correct format

**Sample Input/Output 4:**

Enter the size of an array

**-1**

Array size should be positive

2. Jonas gives an assignment to his students. As per the assignment, he gave a sentence and two numbers as the starting and ending index, and the students had to display the letters from the sentence that were present from the given starting index to the ending index. Help them to complete this assignment using a Java program.

Write a public **class UserInterface**. Write a method **extractString()** as

**public String extractString(String sentence,int number1, int number2)** - This method should print the letters from the sentence which are present in between the starting and ending index

If the numbers are out of range, then it will throw **StringIndexOutOfBoundsException.**

Whether the assignment is done successfully or not, it should concatenate the String as **"Thanks for using the application".**

When the assignment is done successfully, it should return a message "**The output string is <Letter in between the index>.**"

If it results in **StringIndexOutOfBoundsException**, it should return a message, **Invalid index position."**

**Note:**

- "Use try, catch and finally to perform the above task inside **extractString()** method". Write the main method and test the above function.
- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Do not use System.exit(0) to terminate the program.**

**Sample Input/Output 1:**

Enter the String

**Nothing is impossible**

Enter First Index

**2**

Enter Second Index

**12**

The output string is thing is i. Thanks for using the application.

**Sample Input/Output 2:**

Enter the String

**All is well**

Enter First Index

**-5**

Enter Second Index

**20**

Invalid index position. Thanks for using the application.

3. The students of the ECE department at the college have developed a campus radio station and set the frequency range from 91 to 110. Frequency 91.2 MHz is used for music, 93.5 MHz for education, 98.9 MHz for drama, and 109.4 MHz for college events.

If the audience tunes in to the right frequency signal, then the system should display the message "**<input frequency> is the right frequency**". If the audience tunes to a frequency other than the one given in the signal, then the system should alert the audience with the message "**The frequency <input frequency> is out of range**". So, help them automate by writing a Java program that will throw a user defined exception, **"StationNotAvailableException"**.

A partial code is given to do the above task. In the Validator class, you are provided with the method public static boolean **validateStation**(float freq).

The logic for validating the radio station frequency is partially given. Complete the code as follows If the frequency is within the given list, this method should return true. If the frequency is not in the list, this method should throw a "StationNotAvailableException".

From the main method invoke the validateStation method and handle the exception and print the appropriate message.

**Assumption:**

- Assume the given input is in MHz.

**Note:**

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Do not use System.exit(0) to terminate the program.**

**Sample Input and Output 1:**

Scan the radio station

**91.2**

91.2 is the right frequency

**Sample Input and Output 2:**

Scan the radio station

**91.6**

The frequency 91.6 is out of range

4. Betsy bought a dog. She finds it very difficult to find a name for her pet. So, she asked all her friends to suggest a name. The names suggested by her friends are stored in the file. Write a Java program to display the name, which has exactly two vowel alphabets in it (the name can contain other alphabets but only two vowels).

Assume there is only one name that has exactly two vowels (along with other alphabets) in the file.

To do this, create a public class UserInterface with a method findPetName() as follows :

**public String findPetName(String fileName)** - This method should do the following :

- Traverse the input file and find the pet's name that has exactly two vowels, and return that name.
- This program may generate a **FileNotFoundException** or an **IOException**.
- In the case of a **FileNotFoundException**, return a String with the message "**Please give the correct file name**".
- In case of an **IOException**, return a String with the message "**Error occurred while reading the names from file**".
- Use an exception handling mechanism to handle the exception. Use a separate catch block for handling each exception. In the catch block, return the appropriate message.

You are provided with the main method in the UserInterface class. Invoke the method findPetName by passing the file name, and display the name returned by the method findPetName.

**Requirements:**

- Use the try with resources statement inside the method findPetName.
- The **try-with-resources** statement is a try statement that declares one or more resources.
- Use InputFileReader to read the file and BufferedReader to read the file content.
- Declare the resources InputFileReader and BufferedReader inside the try statement.

- When passing the given file to the method findPetName(), the output of the program would be "**Duke**" (Duke has exactly 2 vowels).
- If more than one name has exactly two vowels, return the name that appeared first in the file.

**Note:**

- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Please do not use System.exit(0) to terminate the program**

4. The students of the ECE department at the college have developed a campus radio station and set the frequency range from 91 to 110. Frequency 91.2 MHz is used for music, 93.5 MHz for education, 98.9 MHz for drama, and 109.4 MHz for college events.

If the audience tunes in to the right frequency signal, then the system should display the message "**<input frequency> is the right frequency**". If the audience tunes to a frequency other than the one given in the signal, then the system should alert the audience with the message "**The frequency <input frequency> is out of range**". So, help them automate by writing a Java program that will throw a user defined exception, **"StationNotAvailableException"**.

A partial code is given to do the above task. In the Validator class, you are provided with the method public static boolean **validateStation**(float freq).

The logic for validating the radio station frequency is partially given. Complete the code as follows If the frequency is within the given list, this method should return true. If the frequency is not in the list, this method should throw a "StationNotAvailableException".

From the main method invoke the validateStation method and handle the exception and print the appropriate message.

**Assumption:**

- Assume the given input is in MHz.

**Note:**

- In the Sample Input / Output provided, the highlighted text in bold corresponds to the input given by the user and the rest of the text represents the output.
- Ensure to follow the object-oriented specifications provided in the question.
- Ensure to provide the names for classes, attributes, and methods as specified in the question.
- Adhere to the code template, if provided.

**Do not use System.exit(0) to terminate the program.**

**Sample Input and Output 1:**

Scan the radio station

**91.2**

91.2 is the right frequency

**Sample Input and Output 2:**

Scan the radio station

**91.6**

The frequency 91.6 is out of range