**PROBLEM 1 :**

In the digital city of Numeria, each number has its own unique energy. Some numbers are long and boastful, made up of multiple digits. Others are quiet and small — single digits, often overlooked.

The wise sage of Numeria discovered a hidden pattern:

Some multi-digit numbers are spiritually connected to smaller numbers through the sum of their digits.

For example,

The number 44 is spiritually connected to 8 because 4 + 4 = 8 and 8 also appears in the list.

The number 123 is spiritually connected to 6 because 1 + 2 + 3 = 6

Your task is to help the sage find all such connections.

**Constraints**:

* Input consists of a single line of space-separated integers.
* A number is spiritually connected to another if: The sum of its digits equals a number that also appears in the input.
* If any negative number has appeared as input, then display, **"The input <negative number> is invalid"**and then terminate the program.
* If there is no connection between the numbers then display, **"No match found in the given numbers <input>"**.

**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.
* Adhere to the code template, if provided.
* Please do not use System.exit(x) to terminate the program.

**Sample Input and Output 1:**

Enter the numbers:

**44 123 8 6 11 2**

44 - 8

123 - 6

11 - 2

**Sample Input and Output 2:**

Enter the numbers:

**45 89 92 33 2**

No match found in the given numbers 45 89 92 33 2

**Sample Input and Output 3:**

Enter the numbers:

**44 123 8 6 -5 3 7**

The input -5 is invalid

**PROBLEM 2 :**

Sky Apps is one of the famous shopping applications. They want to calculate the product price based on the product type. Help them as a software developer to do this task.

**Functional Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type (Class)** | **Attribute** | **Methods** | **Responsibilities** |
| **Product** | int productId  String productType  int noOfProduct  double perProductPrice | The getters and setters method for all the attributes and a parameterized constructor of four arguments in the following order- productId, productType, noOfProduct, perProductPrice are provided as a part of the code skeleton. |  |
| **DomesticProduct** | double discount | The getter and setter method for the attribute are provided in the code skeleton.  Include a public 5 argument constructor in the order productId, productType, noOfProduct, perProductPrice, and discount. |  |
| **DomesticProduct** |  | Include the method, **calculateDomesticProductPrice**in the code skeleton. | The method calculates the domestic product price by multiplying the perProductPrice with the noOfProduct, then reducing the discount percentage from the total calculated price.  **Constraint:**   * Return the calculated price as double. |
| **ImportedProduct** | double additionalTax | The getter and setter method for the attribute are provided in the code skeleton.  Include a public 5 argument constructor in the order productId, productType, noOfProduct, perProductPrice, and additionalTax. |  |
| **ImportedProduct** |  | Include the method, **calculateImportedProductPrice**in the code skeleton. | The method calculates the imported product price by multiplying the perProductPrice with the noOfProduct, then add the additionalTax to the total calculated price.  **Constraint:**   * Return the calculated price as double. |

**You are provided with the main method in the UserInterface class as code template, and it is excluded from evaluation.**

**Note:**

* The **DomesticProduct** and **ImportedProduct** are the child classes of the Product class.
* Edit only the **DomesticProduct** and **ImportedProduct** classes to implement the business requirements.
* Assume that all the input values are valid.
* The methods and the constructor should be public, and the attributes of the class should be private.
* 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 that the names for classes, attributes, and methods are provided as specified in the question description.
* **Please do not use System.exit(0); to terminate the program.**

**SAMPLE INPUT AND OUTPUT 1:**

Enter product id

**111**

Enter product type

**DomesticProduct**

Enter number of products

**3**

Enter per product price

**500**

Enter discount percentage

**8**

Total Domestic product price = 1380.0

**SAMPLE INPUT AND OUTPUT 2:**

Enter product id

**333**

Enter product type

**local**

local is an invalid product type

**PROBLEM 3 :**

Rainbow Event Management is one of the famous event management companies in the city. They want to calculate the payment of the artist based on the artist type. Help them as a software developer to do this task.

**Functional Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type (Class)** | **Attribute** | **Methods** | **Responsibilities** |
| **Artist** | int artistId  String artistName  String artistType  int experience | The getters and setters method for all the attributes and a parameterized constructor of four arguments in the following order- **artistId, artistName, artistType, experience** are provided as a part of the code skeleton. |  |
| **Singer** | int noOfSong | The getter and setter method for the attribute are provided in the code skeleton.  Include a public 5 argument constructor in the order - **artistId, artistName, artistType, experience,** and **noOfSong**. |  |
| **Singer** |  | Include the method, **calculatePaymentOfSinger**in the code skeleton. | The method calculates the payment for the singer by multiplying their experience with the noOfSong and with a basic payment of 10000 rupees.  **Constraint:**   * The calculated payment should be returned as double. |
| **Dancer** | int noOfChoreography | The getter and setter method for the attribute are provided in the code skeleton.  Include a public 5 argument constructor in the order - **artistId, artistName, artistType, experience,** and **noOfChoreography**. |  |
| **Dancer** |  | Include the method, **calculatePaymentOfDancer**in the code skeleton. | The method calculates the payment for the dancer by multiplying their experience with the noOfChoreography and with a basic payment of 10000 rupees.  **Constraint:**   * The calculated payment should be returned as double. |

**You are provided with the main method in the UserInterface class as code template, and it is excluded from evaluation.**

**Note:**

* The **Singer** and **Dancer** are the child classes of the Artist class.
* Edit only the **Singer** and **Dancer** classes to implement the business requirements.
* Assume that all the input values are valid.
* The methods and the constructor should be public, and the attributes of the class should be private.
* 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 that the names for classes, attributes, and methods are provided as specified in the question description.
* **Please do not use System.exit(0); to terminate the program.**

**SAMPLE INPUT AND OUTPUT 1:**

Enter Artist id

**123**

Enter Artist name

**Akshara**

Enter Artist type

**Dancer**

Enter years of experience

**8**

Enter number of choreographies done

**20**

Payment of Dancer is 1600000.0

**SAMPLE INPUT AND OUTPUT 2:**

Enter Artist id

**100**

Enter Artist name

**Sara**

Enter Artist type

**writter**

writter is an invalid artist type

**PROBLEM 4 :**

Guru tasked his students with reversing a specific word within a given sentence. Help the students to solve this task using a Java program.

**Constraints:**

* The word to be reversed must be present in the sentence; otherwise, print "**<word to be reversed> is an invalid word**".
* The word to be reversed is case-sensitive.

**Note:**

* Do not edit the existing code template.
* 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.
* Implement the business requirements within the main method. Please do not change the class name.
* **Please do not use System.exit(0); to terminate the program.**

**Sample Input/Output 1**

Enter the sentence

**Do you wear your mask**

Enter the word to reverse

**wear**

Do you raew your mask

**Sample Input/Output 2**

Enter the sentence

**Refer a friend**

Enter the word to reverse

**my**  
  
my is an invalid word

**PROBLEM 5 :**

Pooja and Remya play a word game in their free time, where they find the longest word in a sentence, and then print the resulting string in reverse. Let's implement this game in Java!

**Constraints:**

* The sentence must contain at least 2 words; otherwise, print **"<sentence> is an invalid sentence"** and terminate the program.
* When there is any word in the sentence that does not contain an alphabet, the program should print "***<word> is an invalid word"*** and terminate the program. If there are multiple invalid words, the program should print the first word found in the sentence.
* When more than one word has the same length, then the word that comes first in the sentence is considered.
* Print the output in reverse order.

**Note:**

* Do not edit the existing code template.
* 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.
* Implement the business requirements within the main method. Please do not change the class name.
* **Please do not use System.exit(0); to terminate the program.**

**Sample Input / Output 1**

Enter the sentence

**Praise the lord**

esiarP

**Sample Input / Output 2**

Enter the sentence

**beauti$ful@ w0r|d**

beauti$ful@ is an invalid word

**PROBLEM 6 :**

Once, a language model named Word Wand was tasked with manipulating sentences by a curious user.

Word Wand took the challenge: for odd word counts, it reversed each word; for even word counts, it swapped words in reverse order.

Create a software program to automate this manipulation task.

Constraints:

* The sentence should contain only alphabets and spaces; otherwise, print "Invalid Sentence".

Note:

* Do not edit the existing code template.
* 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.
* Implement the business requirements within the main method. Please do not change the class name.
* Please do not use System.exit(0); to terminate the program.

Sample Input / Output 1:

Enter the sentence

The Sun Shine

Word Count: 3

ehT nuS enihS

Sample Input / Output 2:

Enter the sentence

Dream big work hard and achieve

Word Count: 6

achieve and hard work big Dream

Sample Input / Output 3:

Enter the sentence

The number 42 is often referred to as the answer to life, the universe, and everything

Invalid Sentence

**PROBLEM 7 :**

DD, the host of the new game show "***Broken Telephone***," needs your coding skills to count the number of players who mishear or whisper a secret message by comparing the message with the previous person and the immediate next person. Players stand in a straight line, passing the message from one to the next until it reaches the end. Can you help her automate this process?

***Constraints***

* The first player receives the message on a sheet of paper, so he or she cannot mishear it.
* The messages should be greater than zero, otherwise, display***"<message> is an invalid message"***
* When the messages passed by the players are not similar, count the number of players who misheard or whispered wrongly; otherwise, provide a score of 0.
* The number of players should be greater than or equal to 3, otherwise it will display ***"<number> is an invalid number of players"***.
* If the inputs are valid, go through the messages. For each player, compare their message with the previous and next. Increase the count by 1 if their message doesn't match either. Finally, display "S***core is <count>***".

**Note:**

* Do not edit the existing code template.
* 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.
* Implement the business requirements within the main method. Please do not change the class name.
* **Please do not use System.exit(0); to terminate the program.**

**Sample Input / Output 1**

Enter the number of players

**7**

Enter the messages stated by each player

**1 1 1 3 3 3 2**

Score is 4

**Explanation:**Here, a message is given as input, we have to compare the message with the previous person and the immediate next person. The messages from the **3rd**, **4th**, **6th**, and**7th** players do not match.

**Sample Input / Output 2**

Enter the number of players

**-4**

-4 is an invalid number of players

PROBLEM 8 :

Ram received an assignment from his teacher. He must write Java code for Amazon's and Flipkart's basic flatforms. Assist him in following requirements.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type (Class)** | **Attribute** | **Methods** | **Responsibilities** |
| **OnlineShopping** | String userId  String productName  int quantity  double price | The getters, and setters methods for all the attributes, also no argument constructor are provided as a part of the code skeleton. | Include a parameterized constructor of four arguments in the following OnlineShopping class -*userId,  productName, quantity and price.*  Include the abstract method in the OnlineShopping class  *abstract public double calculatePrice()* |
| **Flipkart** | double superCoins | The getters, and setters methods for all the attributes, also no argument constructor are provided as a part of the code skeleton. | Include a public 5 argument constructor in the order- userId, productName, quantity, price and superCoins. |
| **Flipkart** |  | **calculatePrice** | The method calculates the total price and set the value for supercoins based on the calculated total price.  **Constraints:**   * The total price is calculated as the product of quantity and price. * When the calculated total price is more than 2000, set the *supercoins*by subtracting 2000 from the total price. Otherwise, set the number of *superCoins*to 0. * If quantity or price is less than or equal to 0, then set the value for superCoins as 0. * The calculated total price should be returned as double. |
| **Amazon** | String subscription | The getters, and setters methods for all the attributes, also no argument constructor are provided as a part of the code skeleton. | Include a public 5 argument constructor in the order- *userId, productName, quantity,  price and subscription* |
| **Amazon** |  | **calculatePrice** | The method calculates the total price and sets the value for the subscription based on the calculated total price.  **Constraints**:   * The total price is calculated as the product of quantity and price. * When the total price is greater than or equal to 2000, then set the value for the subscription as "**Congrats, you have got 1 month free subscription**". Otherwise set the value for the subscription  as "**Not eligible for subscription**". * If quantity or price is less than or equal to 0, then set the value for subscription as "***No Subscription***". * The calculated total price should be returned as double. |

**You are provided with the main method in the UserInterface class as a code template, and it is excluded from evaluation.**

**Note:**

* The platform namecan be either "**Flipkart**" or "**Amazon**" (case-insensitive).
* The **Flipkart**and **Amazon**class implements abstract methods of the **OnlineShopping**class.
* Edit only the **Flipkart**and **Amazon**classes to implement the business requirements.
* Assume that all the input values are valid.
* The methods and the constructor should be public, and the attributes of the class should be private.
* 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 that the names for classes, attributes, and methods are provided as specified in the question description.
* **Please do not use System.exit(0); to terminate the program.**

**SAMPLE INPUT AND OUTPUT 1:**

Enter user id

**flp003**

Enter the product name

**Book**

Enter quantity

**2**

Enter price

**1500**

Enter platform name

**Flipkart**

Flipkart:

User id: flp003

ProductName: Book

Quantity: 2

Total price: 3000.0

You have earned  1000 super coins

**SAMPLE INPUT AND OUTPUT 2:**

Enter user id

**Amazon600**

Enter the product name

**Mouse**

Enter quantity

**1**

Enter price

**470**

Enter platform name

**Meesho**

Invalid platform name

**PROBLEM 9 :**

The administration of a college has requested Vixas Tech to develop comprehensive management software for their institution to collect and manage performance evaluations. As a software engineer, your task is to assist in the development and implementation of this project, ensuring the software effectively meets the college's requirements for evaluating and tracking student performance.

|  |  |  |  |
| --- | --- | --- | --- |
| **Type (Class)** | **Attribute** | **Methods** | **Responsibilities** |
| **Person** | int id  String name  String joiningDate  String role | The getters and setters method for all the attributes is provided as a part of the code skeleton. | Include a parameterized constructor of four arguments in the following order- id, name, joiningDate, role.  Include an abstract method in Person class  abstract public int performanceRating() |
| **Student** | double cgpa | The getter and setter method for the attribute are provided in the code skeleton. | Include a public 5 argument constructor in the order- id, name, joiningDate, role and cgpa |
| **Student** |  | **performanceRating** | The method determines the student's performance rating based on their CGPA.   |  |  | | --- | --- | | **cgpa(inclusive)** | **rating** | | 1-3 | 1 | | 4-5 | 2 | | 6-7 | 3 | | 8-9 | 4 | | >9 | 5 |   **Constraints:**   * When the entered CGPA does not match any of the criteria listed in the above table, the method returns -1. Otherwise, the corresponding rating is returned as an integer. |
| **Staff** | int experience | The getter and setter method for the attribute are provided in the code skeleton. | Include a public 5 argument constructor in the order- id, name, joiningDate, role and experience |
| **Staff** |  | **performanceRating** | The method determines the student's performance rating based on their experience.   |  |  | | --- | --- | | **experience (inclusive)** | **rating** | | 1-5 | 1 | | 6-10 | 2 | | 11-15 | 3 | | 16-20 | 4 | | >20 | 5 |   **Constraints:**   * When the entered experience does not match any of the criteria listed in the above table, the method returns -1. Otherwise, the corresponding rating is returned as an integer. |

**You are provided with the main method in the UserInterface class as code template, and it is excluded from evaluation.**

**Note:**

* The role can be either "**Student**" or "**Staff**" (case-sensitive).
* The **Student** and the **Staff** class implements the abstract methods of the Person class
* Edit the Person, Student and Staff classes to implement the business requirements.
* Assume that all the input values are valid.
* The methods and the constructor should be public, and the attributes of the parent class should be protected, and child class should be private.
* 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 that the names for classes, attributes, and methods are provided as specified in the question description.
* **Please do not use System.exit(0); to terminate the program.**

**SAMPLE INPUT AND OUTPUT 1:**

Enter your id

**7834**

Enter your name

**Peter**

Enter the join date

**21-04-2019**

Enter your role

**Student**

Enter your cgpa

**8.1**

Student:

Id: 7834

Name: Peter

Joining date: 21-04-2019

Performance rating: 4

**SAMPLE INPUT AND OUTPUT 2:**

Enter your id

**9274**

Enter your name

**Bharath**

Enter the join date

**14-05-2015**

Enter your role

**Manager**

Invalid role

**PROBLEM 10 :**

Shameera's new camera names photos like "***DSC011***" or "***DSC125001***". "DSC" comes first, followed by the month (01-12) and the file number. Help her rename them with the month, like "JAN\_1" or "DEC\_5001", using Java.

**Constraints:**

* The input string must have minimum of 6 characters and first three characters is 'DSC' always (Case sensitive). Otherwise print **"Invalid Input"**.
* If the input size is valid then check for the month in next two characters. Month should be in between 01 and 12 (both inclusive) otherwise print **"Invalid month"**
* If the month is valid then check for the file number which is from the sixth character. Assume file number always be a numeric value which is greater than zero.

**Note:**

* Do not edit the existing code template.
* 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.
* Implement the business requirements within the main method. Please do not change the class name.
* **Please do not use System.exit(0); to terminate the program.**

**Sample Input / Output 1**

Enter the photo name

DSC0215

Renamed as FEB\_15

**Sample Input / Output 2**

Enter the photo name

DSC000015

Invalid month