BATCHS

**1.Harsha and Prashanti and Manikanta**

The Sprinter sports club conducted a zone level relay running race. Due to the difficult scenario of covid, the Sprinter Sports Club held a zone level relay running competition in a nontraditional way, requiring the teams to compete individually and measuring the time it took each team to complete the course. The team that finishes the race in the shortest amount of time will be declared the winner. Create a Java application to assist the sports club in identifying its winners.

**Assumptions:**

* Assume each team has 4 members
* Take input from the user as a single String with fields separated by colon.The format of the input string is shown below.
* TeamName:TimeTakenByMember1:TimeTakenByMember2:TimeTakenByMember3:TimeTakenByMember4.
* Time taken by each member must be in float

**Requirements:**

* If the number of teams is less than 2 , print**"<numberOfTeams> is an invalid input"**and terminate the program
* If the time taken by each member is less than 1, print **"Invalid number"**and terminate the program
* The output should be like**"<winning team name> team wins the race in <total time taken> minutes"**
* If two teams finish in the same amount of time, the team that started the race first will win.
* Assume that the user inputs will be in the order of the teams that started the race first, second, and so on.
* The output print statement should display the time with a precision of up to two decimal places. Use System.out.printf(""%.2f"", ""time""); for this purpose.

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided.

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

**Sample Input/Output 1:**

Enter the number of teams

**4**

Enter the details

**Green:1.10:1.05:1.00:1.02**

**Red:1.05:1.04:1.10:1.00**

**Blue:1.11:1.10:1.10:1.05**

**Yellow:1.09:1.15:1.08:1.10**

Green team wins the race in 4.17 minutes

**Sample Input/Output 2:**

Enter the number of teams

**-2**

-2 is an invalid input

**Sample Input/Output 3:**

Enter the number of teams

**2**

Enter the details

**White:1.02:1.03:0:1.03**

Invalid number

**Sample Input/Output 4:**

Enter the number of teams

**2**

Enter the details

**Red:1.00:1.00:1.00:1.00**

**White:1.00:1.00:1.00:1.00**

Red team wins the race in 4.00 minutes

**Explanation:**  
Here Red team started the race first, that's why provided as 1st input.  
Since both teams completed the race in same time,  i .e in 4.00 minutes, the team which started the race first will be considered as a winner.  
So, Red team wins the race.

**2. Mandala Lavanya and Gopi**

Guru gave a task to his students. He gave a sentence and the students have to swap the first and the last words and reverse the middle character. Help the students to solve this task using a java program

**Requirements:**

-        The words present in the sentence must be more than 2, else print **"Invalid Length"**

-        The word should contain only alphabets and space, else print **"<sentence> is an invalid sentence"**

**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 description.

-        Ensure to provide the names for classes, attributes, and methods as specified in the question description.

-        Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the sentence

**Do you wear your mask**

mask ruoy raew uoy Do

**Sample Input/Output 2:**

Enter the sentence

**Card reader**

Invalid Length

**Sample Input/Output 3:**

Enter the sentence

**Refer @ friend**

Refer @ friend is an invalid sentence

**3. Meesala Sirisha and Manohar**

SIET private limited is all set to begin a new academic year with a bunch of talents. They need to give employee ids to all the freshers of all team. The employee id is generated by using the training id of the employees which was given at the time of training. Help the company in generating employee id by developing a java application

**Requirements:**

-The training id should  only contain 9 digits, otherwise  print **"<training id> is an invalid training id"** and terminate the application

-The first 4 digits in the training id represent year in which people got trained(it should be 2021), if not print **"<year> is an invalid year"** and terminate the application

-The 5th and 6th digits represent team codes

* 01 - LP
* 02 - TA
* 03 - CT
* 04 - PT
* 05 - TT

-Any other digits than the above mentioned one is given then print **"<code> is an invalid team code"** and terminate the application

-The last 3 digits should be the employees roll number starting from 001 to 999, if not print **"<roll no> is an invalid roll number"** and terminate the application.

-The generated employee id should be 9 characters long with the first 4 letters as **SIET**, the next two letters can be **LP/TA/CT/PT/TT**depending on the team code in the training id and the last 3 digits should be the employee's **roll number.**

**- The trainingId is validated in accordance with the prescribed order.**

**[i.e when 1 or more validation rules gets violated , the first violation message gets displayed. Refer sample output]**

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the training id

**202103069**

Employee Id: SIETCT069

**[Explanation : Employee Id is generated for the valid training id].**

**Sample Input/Output 2:**

Enter the training id

**202115**

202115 is an invalid training id

**[ Explanation: In the given input the length of the training id is less than 9]**

**Sample Input/Output 3:**

Enter the training id

**201005869**

2010 is an invalid year

**[Explanation: In the given input , the year mentioned is incorrect].**

**Sample Input/Output 4:**

Enter the training id

**202119000**

19 is an invalid team code

**[Explanation: In the given input , both team code and the roll number is incorrect].**

**Sample Input/Output 5:**

Enter the training id

**202101000**

000 is an invalid roll number

**[Explanation: In the given input , the roll number is incorrect].**

**4. Kaki Ramya and Sujjith**

The Whale Fishing Club (WFC) conducted a fishing competition on a big lake. The participants allowed for this competition should be 18 years and older. The participants should catch the fishes and give them to the referee so that he will add the count of the fishes based on their size. Then at the end of the day based on the count, points will be awarded. Help WFC by developing an application to convert the count into points in Java.

**Requirements:**

1. There are 3 types of fishes based on their size. Determine points based on the following

Big - 10 points

Medium - 6 points

Small - 3 points

2. Take input from the user as a single string separating each fields with a colon (:)

**<participants name>:<participants age>:<big fish>:<medium fish>:<small fish>**

3. Age should be greater than or equal to 18, else print **"<age> is an invalid age"** and terminate the application

4. If the fish count is less than 0 print **"<fish count> is an invalid input"** and terminate the application

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the details

**Hari:20:5:9:15**

Hari scored 149 points

**Explanation:**

Points calculation

Big fish = 5\*10 = 50

Medium fish =  9\*6 = 54

Small fish = 15\*3 = 45

Total points = 50 + 54 + 45 = 149

**Sample Input/Output 2:**

Enter the details

**Quil:15:2:7:2**

15 is an invalid age

**Sample Input/Output 3:**

Enter the details

**Allan:22:-1:2:3**

-1 is an invalid input

**5. Koyyapu Chandini and Vinay Sai Rahul**

Jack and Joy were playing a game. The game is about one player saying a sentence and the other player has to arrange the letters in that word in alphabetical order. Help Joy to play this game using a java program.

**Requirements:**

* Assume the letters in the sentence is in lowercase.
* The sentence should contain only alphabets and space, else print **"<sentence> is an invalid 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.
* Ensure to follow the object-oriented specifications provided in the question description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the sentence

**the series**

eht eeirss

**Sample Input/Output 2:**

Enter the sentence

**5 is greater than 2**

5 is greater than 2 is an invalid input

**6. Kannepalli Gayatri and Srivali**

Mr. Vicky gave a task to his students. He gave two words, but there were some spelling mistakes found. In the first string, "!" denotes the misspelt word. The second string has the correct spelling of the first string. Create a Java application and help the students remove the "!" and replace it with the correctly spelled character.

**Requirements:**

-        Both the Strings must be of the same length. Otherwise, print**"Length of the strings <String1> and <String2> does not match"**

-        Both the Strings must contain only alphabets and '!' symbol. Otherwise, print**"<String> contains invalid symbols"**

**-**If both the strings contain invalid symbols, print **"<String1> and <String2> contains invalid symbols"**

-        The output must have the combined string without any symbols

Assume that space is allowed in between the words and assume that the second string always will have the correct spelt character of the misspelt first string in the respective positions.

**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 description.

-        Ensure to provide the names for classes, attributes, and methods as specified in the question description.

-        Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the first string

**!ead**

Enter the second string

**rrrr**

read

**Sample Input/Output 2:**

Enter the first string

**F!n! !ut**

Enter the second string

**!i!d O!!**

Find Out

**Sample Input/Output 3:**

Enter the first string

**Po\*\*t\*\*n**

Enter the second String

**!!si!io!**

Po\*\*t\*\*n contains invalid symbols

**Sample Input/Output 4:**

Enter the first string

**!a!i!**

Enter the second string

**V\*l\*d**

V\*l\*d contains invalid symbols

**Sample Input/Output 5:**

Enter the first string

**pr!gra!!i!g**

Enter the second string

**programming**

programming

**Sample Input/Output 6:**

Enter the first string

**C!rr!s**

Enter the second string

**!a!!y**

Length of the strings C!rr!s and !a!!y  does not match

**Sample Input/Output 7:**

Enter the first string

**C%rr\***

Enter the second string

**&a%$y**

C%rr\* and &a%$y contains invalid symbols

**7. Gudla Asritha and Yashash**

Sheethal teaches her little daughter Charu about words and sentences. Sheetal gives Charu a task to find whether the given word is present in the given sentence or not. Help Charu  complete her task by writing a Java program.

If the second input is present in the first input, then print "**<second input> is present in the sentence**". Else print "**<second input> is not present in the sentence**".

**Note:**

* The input word is **case sensitive**
* 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, interface, 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**

**Sample Input1:**

Enter the sentence

**Creativity is always having fun**

Enter the word

**having**

**Sample Output 1:**

having is present in the sentence

**Sample Input2:**

Enter the sentence

**we know one of the most places**

Enter the word

**fun**

**Sample Output2:**

fun is not present in the sentence

**8. Madireddi Monisree and Renetus**

Sundram travels is one of the famous travel agencies, they would like to automate their application. Passengers have a facility to check whether buses are available for their destination. Write a java program to help the travel agency to do this task.

**Assumptions:**

* Assume buses available for these 10 cities only
* Chennai, Coimbatore, Erode, Karur, Madurai, Hyderabad, Salem, Bangalore, Delhi, Agra

**Requirements:**

* If bus available for the city then, print **"Bus for <city name> is available"**
* If bus not available for the city then, print **"Bus for <city name> is not available**"
* City names are case insensitive**.**

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided

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

**Sample Input/Output 1:**

Enter the city name

**Delhi**

Bus for Delhi is available

**Sample Input/Output 2:**

Enter the city name

**Mumbai**

Bus for Mumbai is not available

**Sample Input/Output 3:**

Enter the city name

**chennai**

Bus for chennai is available

**9. Gogada Surya Satya and Meghana**

Digital Technologies has decided to make some updates to the employee profile. So they like to add PAN number and E-mail ID in each profile. Help the firm to validate the PAN number and E-mail ID given by employees using the Java program

**Requirements:**

-        PAN number should have exactly ten characters. The first 5 characters should be alphabets in uppercase, the next four characters should be digits from 0 to 9 and the last character should be an alphabet in uppercase

-        If the PAN number fails the above condition, then print "**<PAN number> is an invalid PAN number**" and terminate the program.

-        E-mail should end with "**@digitec.com"**, before @ there should be at least 5 to maximum 10 characters. All characters should be in lowercase

-        IF E-mail ID fails the above condition, then print "**Invalid E-mail ID**" and terminate the program.

**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 description.

-        Ensure to provide the names for classes, attributes, and methods as specified in the question description.

-        Adhere to the code template, if provided.

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

**Sample Input/Output 1**

Enter your name

**Geetha**

Enter your PAN number

**SSTHP1234L**

Enter your E-mail ID

[**geetha@digitec.com**](mailto:geetha@digitec.com)

Profile of Geetha updated successfully

**Sample Input/Output 2**

Enter your name

**Pradeep**

Enter your PAN number

**QWER8456P**

QWER8456P is an invalid PAN number

**Sample Input/Output 3**

Enter your name

**Sara**

Enter your PAN number

**ABJKL1934F**

Enter your E-mail ID

[**sara@digitec.com**](mailto:geetha@digitec.com)

Invalid E-mail ID

**10. Nocharla Baby Srivani and Manogna**

**Born For Game**, a YouTube channel is conducting a gaming tournament for Rs. 1 Lakh as prize money. They would like to get the player's game code to verify their in-game profiles. So help them to validate the player's code by writing a Java program.

**Requirements:**

* The code should have eight characters. If not print "**<code> does not have the specified length".**
* The code should start with the special character "#". **If not print "<code> does not start with a valid special character".**
* The next 4 characters should be alphabets in upper case. Else print **"<num> is an invalid number for uppercase character".**
* The last 3 characters should be digits from 0 to 9. Else print **"<num> is an invalid number for digits"**.

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided.

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

**Sample Input/Output 1:**

Enter your profile code

**#ACEG321**

#ACEG321 is a valid profile code whose last 3 digits are 321

**Sample Input/Output 2:**

Enter your profile code

**#ZSA8975**

3 is an invalid number for uppercase character

**Sample Input/Output 3:**

Enter your profile code

**QWAE975**

QWAE975 does not start with a valid special character

**Sample Input/Output 4:**

Enter your profile code

**#KKGYI67**

2 is an invalid number for digits

**11. Vyshnavi and Saithi**

.

JammIn this challenge, we use regular expressions (RegEx) to remove instances of words that are repeated more than once, but retain the *first occurrence* of any case-insensitive repeated word. For example, the words love and to are repeated in the sentence I love Love to To tO code. Can you complete the code in the editor so it will turn I love Love to To tO code into I love to code?

To solve this challenge, complete the following three lines:

1. Write a RegEx that will match any repeated word.
2. Complete the second *compile* argument so that the compiled RegEx is case-insensitive.
3. Write the two necessary arguments for *replaceAll* such that each repeated word is replaced with the very first instance the word found in the sentence. It must be the *exact* first occurrence of the word, as the expected output is case-sensitive.

**Note:** This challenge uses a custom checker; you will fail the challenge if you modify anything other than the three locations that the comments direct you to complete. To restore the editor's original stub code, create a new buffer by clicking on the branch icon in the top left of the editor.

**Input Format**

The following input is handled for you the given stub code:

The first line contains an integer, , denoting the number of sentences.  
Each of the  subsequent lines contains a single sentence consisting of English alphabetic letters and whitespace characters.

**Constraints**

* Each sentence consists of *at most*  English alphabetic letters and whitespaces.

**Output Format**

Stub code in the editor prints the sentence modified by the *replaceAll* line to stdout. The modified string must be a modified version of the initial sentence where all repeat occurrences of each word are removed.

**Sample Input**

5

Goodbye bye bye world world world

Sam went went to to to his business

Reya is is the the best player in eye eye game

in inthe

Hello hello Ab aB

**Sample Output**

Goodbye bye world

Sam went to his business

Reya is the best player in eye game

in inthe

Hello Ab

**Explanation**

1. We remove the second occurrence of bye and the second and third occurrences of world from Goodbye bye bye world world world to get Goodbye bye world.
2. We remove the second occurrence of went and the second and third occurrences of to from Sam went went to to to his business to get Sam went to his business.
3. We remove the second occurrence of is, the second occurrence of the, and the second occurrence of eye from Reya is is the the best player in eye eye game to get Reya is the best player in eye game.
4. The sentence in inthe has no repeated words, so we do not modify it.
5. We remove the second occurrence of ab from Hello hello Ab aB to get Hello Ab. It's important to note that our matching is *case-insensitive*, and we specifically retained the *first occurrence* of the matched word in our final string.

**12. Muchinthala Divya and Sreeja**

In this challenge, we use regular expressions (RegEx) to remove instances of words that are repeated more than once, but retain the *first occurrence* of any case-insensitive repeated word. For example, the words love and to are repeated in the sentence I love Love to To tO code. Can you complete the code in the editor so it will turn I love Love to To tO code into I love to code?

To solve this challenge, complete the following three lines:

1. Write a RegEx that will match any repeated word.
2. Complete the second *compile* argument so that the compiled RegEx is case-insensitive.
3. Write the two necessary arguments for *replaceAll* such that each repeated word is replaced with the very first instance the word found in the sentence. It must be the *exact* first occurrence of the word, as the expected output is case-sensitive.

**Note:** This challenge uses a custom checker; you will fail the challenge if you modify anything other than the three locations that the comments direct you to complete. To restore the editor's original stub code, create a new buffer by clicking on the branch icon in the top left of the editor.

**Input Format**

The following input is handled for you the given stub code:

The first line contains an integer, , denoting the number of sentences.  
Each of the  subsequent lines contains a single sentence consisting of English alphabetic letters and whitespace characters.

**Constraints**

* Each sentence consists of *at most*  English alphabetic letters and whitespaces.

**Output Format**

Stub code in the editor prints the sentence modified by the *replaceAll* line to stdout. The modified string must be a modified version of the initial sentence where all repeat occurrences of each word are removed.

**Sample Input**

5

Goodbye bye bye world world world

Sam went went to to to his business

Reya is is the the best player in eye eye game

in inthe

Hello hello Ab aB

**Sample Output**

Goodbye bye world

Sam went to his business

Reya is the best player in eye game

in inthe

Hello Ab

**Explanation**

1. We remove the second occurrence of bye and the second and third occurrences of world from Goodbye bye bye world world world to get Goodbye bye world.
2. We remove the second occurrence of went and the second and third occurrences of to from Sam went went to to to his business to get Sam went to his business.
3. We remove the second occurrence of is, the second occurrence of the, and the second occurrence of eye from Reya is is the the best player in eye eye game to get Reya is the best player in eye game.
4. The sentence in inthe has no repeated words, so we do not modify it.
5. We remove the second occurrence of ab from Hello hello Ab aB to get Hello Ab. It's important to note that our matching is *case-insensitive*, and we specifically retained the *first occurrence* of the matched word in our final string.

**13. Sindhu and Sankya**

Rahul is a 15 years old boy. He bought  a new laptop.  He likes to  password protect his laptop,  So help Rahul to generate a password for his laptop and validate the password by writing a Java program.

**Requirements:**

Hint to generate a password:

* Minimum of 6 characters and a maximum of 12 characters (characters include **alphabets,** **numbers and special characters**).
* The password should contain atleast one uppercase, one lowercase,  and one special character
* Only these special characters @,$,\*,# are allowed
* keeping **numbers** in the password is **optional**.
* If the password fails the above condition, print "**<password> is an** i**nvalid password**" and terminate the program

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided.

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

**Sample Input/Output 1**

Generate your password

**#001Lord**

The generated password **#**001Lordis valid and has 3 lowercase alphabet 1 uppercase alphabet 1 special character 3 digit

**Sample Input/Output 2**

Generate your password

**LIVE@123**

LIVE@123 is an invalid password

**14.Akshay and Sneha**

Sharpx private company is conducting a conference, for which the participants need to pre-register. After completing the registration they will get an ID. By using the ID they can enter the conference hall. So help Sharpx private company to validate the ID issued to the participants.

**Requirements:**

* The ID should have exactly ten characters.
* The format of the ID is "**SPC00105PM**".
* The first 3 characters of the ID should be "**SPC**", which determines the company name.
* The next 3 characters should be digits, it determines the seat number.
* The last 4 characters should determine the time of the conference.
* Of the last four characters, the first 2 digits indicate time, it should be greater than 0 and less than or equal to 12.
* The next two characters are time zone. It should be either "**AM**" or "**PM**".
* If the input does not match with the above condition, print "**<ID> is an Invalid** **ID**" and terminate the program.

**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 description.
* Ensure to provide the names for classes, attributes, and methods as specified in the question description.
* Adhere to the code template, if provided.

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

**Sample Input/Output 1:**

Enter your name

**Aravind**

Enter your ID

**SPC01004PM**

Hi Aravind your seat number is 10 and the event starts at 4PM

**Sample Input/Output 2:**

Enter your name

**Snekha**

Enter your ID

**SPA01004PM**

SPA01004PM is an Invalid ID

**Sample Input/Output 3:**

Enter your name

**Jin**

Enter your ID

**SPC01004HM**

SPC01004HM is an Invalid ID

**15.Ruchitha and Fathima**

Write a class called *MyRegex* which will contain a string pattern. You need to write a regular expression and assign it to the pattern such that it can be used to validate an IP address. Use the following definition of an IP address:

IP address is a string in the form "A.B.C.D", where the value of A, B, C, and D may range from 0 to 255. Leading zeros are allowed. The length of A, B, C, or D can't be greater than 3.

Some valid IP address:

000.12.12.034

121.234.12.12

23.45.12.56

Some invalid IP address:

000.12.234.23.23

666.666.23.23

.213.123.23.32

23.45.22.32.

I.Am.not.an.ip

In this problem you will be provided strings containing any combination of ASCII characters. You have to write a regular expression to find the valid IPs.

Just write the MyRegex class which contains a String . The string should contain the correct regular expression.

(MyRegex class *MUST NOT* be public)

**Sample Input**

000.12.12.034

121.234.12.12

23.45.12.56

00.12.123.123123.123

122.23

Hello.IP

**Sample Output**

true

true

true

false

false

false

**16.Navya Sri and Supraja**

Using **Regex**, we can easily match or search for patterns in a text. Before searching for a pattern, we have to specify one using some well-defined syntax.

In this problem, you are given a pattern. You have to check whether the syntax of the given pattern is valid.

**Note**: In this problem, a regex is only valid if you can compile it using the [Pattern.compile](http://docs.oracle.com/javase/6/docs/api/java/util/regex/Pattern.html" \l "compile%28java.lang.String%29) method.

**Input Format**

The first line of input contains an integer , denoting the number of test cases. The next  lines contain a string of any printable characters representing the pattern of a regex.

**Output Format**

For each test case, print Valid if the syntax of the given pattern is correct. Otherwise, print Invalid. Do not print the quotes.

**Sample Input**

3

([A-Z])(.+)

[AZ[a-z](a-z)

batcatpat(nat

**Sample Output**

Valid

Invalid

Invalid

**17.Prathuysha and G.Divya**

Given a string, , matching the regular expression [A-Za-z !,?.\_'@]+, split the string into *tokens*. We define a token to be one or more consecutive English alphabetic letters. Then, print the number of tokens, followed by each token on a new line.

**Note:** You may find the [String.split](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html" \l "split-java.lang.String-) method helpful in completing this challenge.

**Input Format**

A single string, .

**Constraints**

* is composed of *any* of the following: English alphabetic letters, blank spaces, exclamation points (!), commas (,), question marks (?), periods (.), underscores (\_), apostrophes ('), and at symbols (@).

**Output Format**

On the first line, print an integer, , denoting the number of tokens in string  (they *do not* need to be unique). Next, print each of the  tokens on a new line in the same order as they appear in input string .

**Sample Input**

He is a very very good boy, isn't he?

**Sample Output**

10

He

is

a

very

very

good

boy

isn

t

he

**Explanation**

We consider a token to be a contiguous segment of alphabetic characters. There are a total of  such tokens in string , and each token is printed in the same order in which it appears in string .