2000 ACM South Central Regional Programming Contest

Louisiana State University

Problem #1: Is this poison?

Introduction

You have recently landed a job with an supercomputing application hosting firm. The firm's primary mission is to provide distributed computing for its customers across its networked supercomputer. The supercomputer consists of many individual computers of varying size (i.e. Mainframes, Servers, and Workstations) and configurations (i.e. OS's, hardware, MIPS). Each of these computers are commonly referred to as processing nodes. Customers submit to a centralized distribution point the programs they want executed and the sets of data files to be processed. Each program is distributed to the processing nodes along with a particular data file. The program is then executed on each node with general security privileges. The types of programs that customers submit are thread intensive, utilize a large amount of inter-process communication, and execute generic OS system calls. The results of the processing are then communicated to the submitter.

The firm's customers are primarily competing bioengineering firms researching the human genome. Each is racing to patent particular genes and gene interactions which they discover. Since the applications are submitted directly by the customers and many customers will be submitting programs for distributed computation at the same time, security of the processing nodes and of individual customer data and source code is of paramount importance.

Your project lead is concerned that some customers may try to interject virus or malicious code into the distribution process, either to crash the processing nodes, steal the source code and data of other customers, or affect the programs of other customers in order to produce erroneous data.

You have been tasked with writing a malicious code scanning program for the distribution subsystem. Its primary functional requirement is to scan programs and data files submitted by customers and compare them to the malicious code library your company maintains. This program will produce reports indicating if malicious code exists in the submitted program and data files.

Here are a the requirements given to you by your lead:

- Malicious code characters always appear in files sequentially without interruption i.e. "the dog" will always appear as "the dog" never as "the cat dog".
- A malicious code character may differ from the signature character, however they will only differ in the character ASCII value not in the amount of characters i.e. "the dog" may be represented as "The dog" or " he dog" or "tHe doc" but not as "the ddog". **SPECIAL CASE: at no time will an EOLN (end of line)/ carriage return character replace an ASCII value of a malicious code character.**
- Malicious code match percentages are determined with the following formula:

  Code match percentage = # of characters matching the virus signature

# of characters in the virus signature.

- Code, or data files, are considered CLEAN or POISON.
- "CLEAN" indicates the file does not contain any malicious code matches equaling or exceeding the threshold percentage for any signature. "POISON" indicates that the file does contain malicious code matches exceeding the threshold percentage for at least one signature.
- If more than one instance of malicious code exists, examine them all. If any one exceeds the threshold the code is considered POISON.

Input

Input will consist of up to 100 data sets. Each data set will be correct and follow the following rules:

- A set of malicious code descriptions contains the following:

Elements of a Malicious code description	**Description**	**Example**
Malicious code signatures start line	A set of characters of the form MALICIOUS CODE SIGNATURES followed by a new line character.	`MALICIOUS CODE SIGNATURES`
Any number of Code Signatures	Next bullet in the list	

- Code Signature(s) - 0..n instances of the following:

Elements of a Signature	**Description**	**Example**
Signature description start line	A set of characters of the form <*x*> (any combination of the following valid characters "a".."z", "A".."Z", "0".."9", & ".") followed by a new line character. <*x*> is the name of the signature.	`RABBIT`
Threshold percentage start line	A set of characters of the form THRESHOLD PERCENTAGE followed by a new line character.	`THRESHOLD PERCENTAGE`
Threshold percentage	A set of characters of the form <*y*> (any combination of the following valid characters "0".."9" & ".") followed by a new line character. <y> is the threshold percentage to 2 decimal places of precision i.e. 100.00, 95.03 etc.	`50.00`
Threshold percentage end line	A set of characters of the form THRESHOLD PERCENTAGE followed by a new line character.	`THRESHOLD PERCENTAGE`
Code Signature start line	A set of characters of the form SIGNATURE followed by a new line character.	`SIGNATURE`
Code signature	A set of characters (1..n - all ASCII characters except EOF and EOLN are valid) followed by a new line character. The entire contents of these lines is the code signature.	`We`

Code Signature end line	A set of characters of the form SIGNATURE followed by a new line character.	`SIGNATURE`
Signature description end line	A set of characters of the form SIGNATURE followed by a new line character.	`RABBIT`

Elements of a Malicious code description	Description	Example
Malicious code signatures end line	A set of characters of the form MALICIOUS CODE SIGNATURES followed by a new line character.	`MALICIOUS CODE SIGNATURES`

- Code or data to scan - 1 instance of the following:

Elements of code file to scan	Description	Example
File start line	A set of characters of the form <*z*> (any combination of the following valid characters "a".."z", "A".."Z", "0".."9", & ".") followed by a new line character. <*z*> is the name of the file.	`XCHROMOSOMECALC1`
Data start line	A set of characters of the form DATA followed by a new line character.	`DATA`
Data	A set of characters (1..n - all ASCII characters except EOF are valid) followed by a new line character. The entire contents of these lines is the data or program file to be scanned.	`Class calc {` `public calc() {` `Wefg6me3to 632edcM` `regsdfal p3455agdisg` `fogiest.` `System.out.println("cmo");` `While heM; egimm;` `ing; System` `.out.println("con\ntest");` `}` `}`
Data end line	A set of characters of the form DATA followed by a new line character.	`DATA`
File end line	A set of characters of the form <*z*> (any combination of the following valid characters "a".."z", "A".."Z", "0".."9", & ".") followed by a new line character. <*z*> is the name of the file.	`XCHROMOSOMECALC1`

- Blank line - separating the start of another data set from the previous data set.

Output

For each data set, output will consist of a set of characters of the form "*<z> <r>*" where <*z*> is the name of the data file which was scanned (any combination of the following valid characters "a".."z", "A".."Z", "0".."9", & ".") and <*r*> is a set of characters of the form "CLEAN" or "POISON" followed by a new line character. "CLEAN" indicates the file does not contain any malicious code matches equaling or exceeding the threshold percentage for any signature. "POISON" indicates that the file does contain malicious code matches exceeding the threshold percentage for at least one signature.

Sample Input

```
MALICIOUS CODE SIGNATURES
MALICIOUS CODE SIGNATURES
XCHROMOSOMECALC1
DATA

Class calc {
public calc() {
Wefg6me3to 632edcM
regsdfal p3455agdisg fogiest.
System.out.println("cmo");
While heM; egimm;
ing; System .out.println("con\ntest");
}
}
DATA
XCHROMOSOMECALC1

MALICIOUS CODE SIGNATURES
RABBIT
THRESHOLD PERCENTAGE
50.00
THRESHOLD PERCENTAGE
SIGNATURE
We
SIGNATURE
RABBIT
MALICIOUS CODE SIGNATURES
XCHROMOSOMECALC1
DATA

Class calc {
public calc() {
Wefg6me3to 632edcM
regsdfal p3455agdisg fogiest.
System.out.println("cmo");
While heM; egimm;
ing; System .out.println("con\ntest");
}
}
DATA
XCHROMOSOMECALC1

MALICIOUS CODE SIGNATURES
RABBIT
THRESHOLD PERCENTAGE
70.00
THRESHOLD PERCENTAGE
SIGNATURE
Welcome to the ACM regional programming contest.
```

```
SIGNATURE
RABBIT
FROG
THRESHOLD PERCENTAGE
85.05
THRESHOLD PERCENTAGE
SIGNATURE
Thats the way uh-hu uh-hu I like it..
SIGNATURE
FROG
MALICIOUS CODE SIGNATURES
PROTEINFOLDING.2
DATA

Public Class {
format c:
Thats the ACM regional hu I like it..
}
DATA
PROTEINFOLDING.2
```

Sample Output

```
XCHROMOSOMECALC1 CLEAN
XCHROMOSOMECALC1 POISON
PROTEINFOLDING.2 CLEAN
```