Problem 1

Year of Astronomy

2 points

JAVA: program name must be prob01.java
C /C++ program name must be: prob01.exe

Task Description

In 1609 Galileo turned his telescope up to the night sky and observed moons in orbit around Jupiter and craters on Earth's moon. What he saw (and wrote) revolutionized science and the rest of the world as well. In commemoration of the 400 years since his first observations, the year 2009 has been designated the International Year of Astronomy. You can join the fun by writing a program to calculate attributes of a telescope from its basic measurements.

A telescope is typically composed of two optical elements: the primary or objective, which is usually a large lens or curved mirror, and the eyepiece. The objective is permanently mounted in a tube or truss. Eyepieces, however, are fitted into a focusing tube and can be easily swapped to provide different levels of magnification.

The focal ratio f of a telescope is calculated by dividing the focal length of the objective FO by its diameter D. The exit pupil P (the size of the image that exits the eyepiece) is calculated by dividing the focal length of the eyepiece Fe by the focal ratio f.

```
            f = FO / D
            P = Fe / f
```

Sample Input

The input will contain three integers: the focal length of an objective, a diameter, and focal length of an eyepiece, in that order.

```
1200
250
28
```

Sample Output

The program must calculate the focal ratio and the exit pupil from the input values. The program should print these values on two separate lines, focal ratio first then exit pupil.

```
4.8
5.8333333
```

Problem 2

Polite Numbers

3 points

JAVA: program name must be prob02.java
C /C++ program name must be: prob02.exe

Task Description

A polite number is a positive integer which can be written as the sum of two or more consecutive positive integers. Other positive integers are impolite.

In years past the Code Wars problem team might have left you to fend for yourself to figure out an algorithm for determining whether a number is a power of two. This year, however, we recognize that the stress of the economic recession is already making life difficult for everyone, so here's a hint: impolite numbers are all exactly powers of two.

For example the numbers 3, 6, 20, and 11 are all polite. On the other hand, the numbers 16 and 128 are impolite because:

$$16 = 2^4 = 2 * 2 * 2 * 2$$
$$64 = 2^6 = 2 * 2 * 2 * 2 * 2 * 2$$

Program Input

The input is a single positive integer value.

```
4
```

Program Output

The program must print the input value, followed by a message that says if the number is polite or impolite.

```
4 is impolite
```

Problem 3

Diamonds are Forever

3 points

JAVA: program name must be prob03.java
C /C++ program name must be: prob03.exe

Task Description

This mission, should you decide to accept it, involves drawing diamonds. Should be a simple coding assignment, no?

Impossible? No!

The inputs to your program are the character to use to draw the diamond, and the scale of the diamond. In this mission, scale is the height above and below the midline, including the midline.

Sample Input/Output

```
Character: a
Scale: 5

    a
   a a
  a   a
 a     a
a       a
 a     a
  a   a
   a a
    a
Done
```

Problem 4

Dirt Simple Calculator

4 points

JAVA: program name must be prob04.java
C /C++ program name must be: prob04.exe

Task Description

In years past, the CodeWars problem development team (TM) has designed a number of calculator programs that were, frankly, just way too complicated to be in the contest. This year, in an effort to make amends for the mistakes of the past, the team has provided this dirt simple calculator.

Write a program that functions as a simple calculator.

Input

Each line of the input is a series of integers separated by the operators '+', '-', or '*' and terminated by an equal sign '='. The program should continue to read input lines until it reads a line with only "0 =" on it.

Output

For each line of input, the program should print the result of the operations performed in order from left to right. The multiplication operation does not take precedence over addition and subtraction.

Sample Input

```
28 - 7 * 3 =
13 * 4 + 8 * 2 + 1 =
4 + 3 * 52 =
0 =
```

Sample Output

```
63
121
364
```

Problem 5

Going Green

4 points

JAVA: program name must be prob05.java
C /C++ program name must be: prob05.exe

Task Description

A research team has installed CO_2 monitoring equipment in a variety of locations around the globe. Data from these monitors is stored in a central database and the team produces reports on a monthly basis.

Write a program to find peak CO_2 levels in the measured data.

A peak is defined as a value that is greater than the value before it and the value after it..

Input

The input will contain multiple data sets. Each data set consists of two lines. The first line gives the month and year that the data was collected. The next line consists of a series of integers that represent the measured amount of CO_2 for a particular day. There will be one measurement reported per day. The input ends with a line that says "END 0".

Output

Your program must find peak CO_2 levels in the data. For each peak the program must print the date of the peak and the actual CO_2 measurement. If a peak spans two or more days, the program must print the entire date range of the peak.

Sample Input

```
JUNE 2003
48 52 47 46 44 42 43 44 47 49 44 43 39 40 40 36 31 36 36 37 42 51 59 62 62 67 75 70 66 69
FEBRUARY 2005
66 70 73 80 83 84 86 82 87 88 88 90 89 95 98 100 100 100 89 88 87 86 85 85 85 88 88 84
END 0
```

Sample Output

```
JUNE 2, 2003: 52
JUNE 10, 2003: 49
JUNE 14-15, 2003: 40
JUNE 27, 2003: 75
FEBRUARY 7, 2005: 86
FEBRUARY 12, 2005: 90
FEBRUARY 16-18, 2005: 100
FEBRUARY 26-27, 2005: 88
```

Problem 6

Kepler's Anomaly

5 points

JAVA: program name must be prob06.java
C /C++ program name must be: prob06.exe

Task Description

In the year 1605 Johannes Kepler discovered three basic laws of planetary motion. These laws are still used today for many applications where the positions of the planets need to be calculated. The first law states that a planet's orbit is not a circle but an ellipse with sun at one focus. The calculation of a planet's position require the use of variables with fancy names like "eccentricity" and "mean anomaly".

Luckily for you, for this program you won't need a degree in astrophysics. You don't even need to know what "eccentric anomaly" actually means. You will, however, have to write a program that can use iterations (in other words, a loop) to calculate a numeric value from a formula.

The eccentric anomaly E of a planet can be calculated from its eccentricity e and its mean anomaly M. Starting with an approximation E1 for E, a better approximation E2 can be calculated with this formula:

```
E2 = M + e * sin(E1)
```

The final value of E is computed by iterating (looping) the formula until the difference between E1 and E2 is less than a desired value v.

In Java and C++ the sin function expects an angle in radians, so all input and output angles will be in radians.

Sample Input

Each line of the input will contain the name of a planet, followed by four numbers: the values for M, e, E1, and v. The input ends with the word END and four zeros.

```
MERCURY 1.2 0.205635 1.391660 0.00001
VENUS 3.384683 0.0067547 4.18880 0.00001
PLUTO 4.170094 0.248808 4.660029 0.00001
END 0 0 0 0
```

Sample Output

For each planet the program must calculate the value of E until the absolute value of E2 - E1 is less than the value v. The program must then print the name of the planet and the final value for the eccentric anomaly.

```
MERCURY 1.40274
VENUS 3.38307
PLUTO 3.98436
```

Problem 7

Ponzi Calculator

5 points

JAVA: program name must be prob07.java
C /C++ program name must be: prob07.exe

Task Description

You've decided to retire early using a fool-proof pyramid, or "Ponzi" scheme. A Ponzi scheme is a fraudulent investment proposal that uses money collected from new investors (victims) to pay off previous investors. The problem with a Ponzi scheme is that eventually, there just aren't enough new investors to pay off all the existing ones, and the pyramid collapses.

For your scheme, you're going to tell investors about the wonders of Emu farming. You promise to raise the Emus at a "secret farm in Montana", and sell their wool to high-end clothing makers. You'll pay them interest every month and investors will quickly double their money or more!

You need $10,000,000 to buy a beach house in the Caymen Islands. Write a program to figure out how much to steal, err, borrow, from each investor, how many investors each month you can swindle, and what interest rate will intice them to invest. When you have enough cash, you'll skip town. Simple!

The Annual Interest Rate will be a whole integer between 1-20. For simplicity, assume that each month is exactly 1/12 of a year, all new investors come in at the beginning of the month, and all interest payments are made at the end of the month.

Input

Prompt for the dollar amount each "valued investor" will contribute (1-1000000), the number of new investors you'll recruit each month (1-20000), and the Annual Interest Rate (1-400) you'll promise everyone. For simplicity, assume that each month is exactly 1/12 of a year, all new investors come in at the beginning of the month, and all payments are made at the end of the month.

Output

The program will output the balance sheet at the end of the month. For each month, print the number of investors, the account balance once once new investments are made, the interest to be paid at the end of the month, and the balance at the end of the month. If the balance before paying interest is over $10,000,000, print the total balance, the number of months that have passed, and the words "Purple Horseshoe loves Emu Farms!" - your secret phrase to let you know it's time to run for the airport. If at any time the balance goes below zero, the gig is up, so print out "Abort Project Emu Farms!".

Sample Inputs/Outputs

```
Investment Cost (per person) $:  5000
New Investors (per month):  500
Guaranteed Annual Interest Rate (%):  180

Month Investors Begin Balance  Interest Paid  End Balance
----- --------- -------------- -------------- ---------------
1         500       $2500000     $375000          $2125000
2         1000      $4625000     $750000          $3875000
3         1500      $6375000     $1125000         $5250000
4         2000      $7750000     $1500000         $6250000
5         2500      $8750000     $1875000         $6875000
6         3000      $9375000     $2250000         $7125000
7         3500      $9625000     $2625000         $7000000
8         4000      $9500000     $3000000         $6500000
9         4500      $9000000     $3375000         $5625000
10        5000      $8125000     $3750000         $4375000
11        5500      $6875000     $4125000         $2750000
12        6000      $5250000     $4500000         $750000
13        6500      $3250000     $4875000         -$1625000

Abort Project Emu Farms!
```

```
Investment Cost (per person) $:  5000
New Investors (per month):  500
Guaranteed Annual Interest Rate (%):  150

Month Investors Begin Balance  Interest Paid  End Balance
----- --------- -------------- -------------- ---------------
1         500       $2500000     $312500          $2,187500
2         1000      $4687500     $625000          $4,062500
3         1500      $6562500     $937500          $5,625000
4         2000      $8125000     $1250000         $6,875000
5         2500      $9375000     $1562500         $7,812500
6         3000      $10312500    $1875000         $8,437500

Total Balance at beginning of Month 6 is $10312500
Purple Horseshow loves Emu Farms!
```

Problem 8

Security Door

6 points

JAVA: program name must be prob08.java
C /C++ program name must be: prob08.exe

Task Description

Sid the intern has completed the initial wiring of a lab network in between sessions of eating snacks and texting his friends about cars and girls. The network connects via fiber optic cable to equipment in a wiring closet. Sid worries that one of the other interns could walk in and accidently disconnect the cable. He's seen this sort of thing before. Although Sid looks quite intimidating in his red robot boots and animé hero blue hair, he knows he can't guard the closet 24/7.

To keep the equipment secure, Sid the intern has devised a security system with an electronic door lock that can only be opened using a wireless remote. With the press of a button the remote transmits a code to a receiver inside the closet. The receiver verifies the code and opens the lock if it is valid. For added security the code changes each time the button is pressed according to this formula, known as a linear conguential generator:

```
          NewCode = ( 17 * OldCode + 91 ) modulus 256
```

Help Sid write a program for the receiver. After each valid code is received, the receiver must recognize as valid the next three codes in the sequence just in case Sid accidently presses the button while he's away from the door.

Input

The input will consist of a series of command-integer pairs. The SYNC command tells the receiver to use the following number as its "Old Code". The OPEN command requests the door to open using the following number as the security code. The input ends with the command END 0.

Output

When the program reads a SYNC command it must print the word SYNCHRONIZED. When the program reads an OPN command with a valid code it must print the word UNLOCK. If it reads an OPEN command with an invalid code it must print the word INVALID.

Sample Input

```
SYNC 37
OPEN 54
OPEN 241
OPEN 133
OPEN 66
SYNC 63
OPEN 133
END 0
```

Sample Output

```
SYNCHRONIZED
UNLOCK
UNLOCK
INVALID
UNLOCK
SYNCHRONIZED
UNLOCK
```

Problem 9

Fun With Palindromes

6 points

JAVA: program name must be prob09.java
C /C++ program name must be: prob09.exe

Task Description

We all know about palindromes. "Bob" is a simple palindrome. Did you know that you can generate numeric palindromes using a simple algorithm?

Algorithm to find numeric palindromes:

1. Input a number. Call it num1.
2. Set num2 equal to the reverse of num1.
3. Check to see if num1 equals num2. If so, output the number. Stop
4. Set num1 = num1 + num2.
5. Go to Step 2.

Here is an example:

```
181
    1.  num1 = 181
    2.  num2 = 181
    3.  num1 equals num2. Stop
```

It took 0 complete traversals of the algorithm to find the numeric palindrome of 181. Total number of steps = 0

```
90
    1.  num1 = 90
    2.  num2 = 9
    3.  num1 does not equal num2
    4.  num1 = 90 + 9

Start at Step 2.
    2.  num2 = 99
    3.  num1 equals num2,  Stop
```

It took one full traversal of the algorithm to find the numeric palindrome of 90 (99). Total number of steps = 1.

Your task is to compute numeric palindromes for a given range of numbers.

Input

<Starting number> <Ending number>

Output

The number, number of steps used to calculate the palindrome, and the numeric palindrome.

Sample Input/Output

```
>> palindromes 929 936
Finding Palindromes for numbers from 929 to 936...
Number of Palindromes calculated: 8

|No.|Steps|Palindrome
|---|-----|--------------
|929|    0|929
|930|    1|969
|931|    2|1771
|932|    2|2882
|933|    2|3993
|934|    3|9119
|935|    4|25652
|936|    5|99099
```

Problem 10

Data in a Huff

6 points

JAVA: program name must be prob10.java
C /C++ program name must be: prob10.exe

Task Description

Compression algorithms are categorized according to whether they allow the exact original data to be reconstructed from the compressed data. Formats like MP3 and JPEG are “lossy”, while formats like ZIP and PNG are “lossless”. Not only can compressed data files be transmitted (across the internet or onto your MP3 player) faster than uncompressed data files, but they also require less storage space.

`' '`	`00`
`'.'`	`010010`
`'!'`	`01011101000`
`','`	`01011110`
`'?'`	`0101111100100`
`'0'`	`010111011`
`'1'`	`110111100`
`'2'`	`010111000`
`'3'`	`11011110111`
`'4'`	`010111110011`
`'5'`	`0101110101`
`'6'`	`11011110100`
`'7'`	`01011101001`
`'8'`	`11011110101`
`'9'`	`0101111100101`
`'A'`	`1000`
`'B'`	`110110`
`'C'`	`10110`
`'D'`	`11010`
`'E'`	`1111`
`'F'`	`01010`
`'G'`	`010110`
`'H'`	`10101`
`'I'`	`0110`
`'J'`	`01011111000`
`'K'`	`010111111`
`'L'`	`10100`
`'M'`	`01000`
`'N'`	`0111`
`'O'`	`1001`
`'P'`	`010011`
`'Q'`	`010111001`
`'R'`	`11100`
`'S'`	`11101`
`'T'`	`1100`
`'U'`	`101111`
`'V'`	`11011111`
`'W'`	`1101110`
`'X'`	`0101111101`
`'Y'`	`101110`
`'Z'`	`11011110110`

Table 1

Huffman coding is a lossless compression algorithm that encodes each input character into a variable-length bit string. Write a program to decode a bit string using the predefined Huffman encoding shown in Table 1.

Sample Input

The input will be a simple string of zeros and ones terminated by a single period.

```
10110100111010111100
11011101000111001110
10011100101111101001
1111110101011101000.
```

Sample Output

The program must print the decoded message to the console. In addition, the program must print the number of bits in the message and the number of bits in the compressed bit string. For this program we define each output character as an eight-bit byte. The numbers must be labeled correctly.

```
CODE WARS RULES!
message bits: 128
compressed bits: 79
```

Problem 11

Mass Intelligence

6 points

JAVA: program name must be prob11.java
C /C++ program name must be: prob11.exe

Task Description

As computers continue to increase in speed and memory capacity, new software will allow us humans to use our machines to solve problems in ways never before possible. The key to making this new software work will be its ability to mimic human intelligence. One important aspect of human intelligence is the manipulation of symbolic representations of knowledge for a field of study.

For example, in chemistry each element is represented by a one or two letter symbol: H for Hydrogen, C for Carbon, Na for Sodium. When several elements have been combined to form a compound, it is represented by a formula in which each element appears. If the compound is made of more than one of a certain element, then that element's symbol is followed by an integer that indicates how many of that element are in a single molecule of the compound.

Furthermore, each element has an atomic mass that indicates the relative mass of one atom of that element. When elements are combined to form a compound, its molecular mass is the sum of the atomic masses of all its constituent elements.

Write a program that can read a chemical formula and determine the molecular mass of the compound.

Table of Atomic Masses

H	1.008
He	4.003
Li	6.941
Be	9.012
B	10.81
C	12.01
N	14.01
O	16.00
F	19.00
Ne	20.18
Na	22.99
Mg	24.31
Al	26.98
Si	28.09
P	30.97
S	32.07
Cl	35.45
Ar	39.95

Sample Input

Each line of the input file is a chemical formula. The input ends with the word END.

```
O2
H2O
CH4
Na2SO4
C6H12O6
BeSO3
NaHCO3
END
```

Sample Output

Your program must print the chemical formula, followed by the molecular mass of the compound described by the formula.

```
O2 32
H2O 18.016
CH4 16.042
Na2SO4 142.05
C6H12O6 180.156
BeSO3 89.082
NaHCO3 84.008
```

Problem 12

Histogram

6 points

JAVA: program name must be prob12.java
C /C++ program name must be: prob12.exe

Task Description

Write a program to count the number of times words appear in the input and print a histogram of the top five most frequent words in order of frequency.

Sample Input

The input will contain several lines of text. The input is terminated by a line with the string "###".

```
        When in the Course of human events, it becomes necessary for one
            people to dissolve the political bands which have connected them
            with another, and to assume among the powers of the earth, the
            separate and equal station to which the Laws of Nature and of
            Nature's God entitle them, a decent respect to the opinions of
            mankind requires that they should declare the causes which impel
            them to the separation.
        ###
```

Sample Output

The program must display a histogram of the top five most frequent words in the input text. Lower-case letters should be converted to upper-case. Punctuation marks should be ignored. Each line of output must contain, in order, the histogram, the rank of a word, the word itself, and the number of occurances. Words with the same frequency must be printed in alphabetical order. Please follow the format of the sample output below.

```
********* #1: THE - 9
***** #2: OF - 5
***** #3: TO - 5
*** #4: AND - 3
*** #5: THEM - 3
```

Problem 13

IP Router

7 points

JAVA: program name must be prob13.java
C /C++ program name must be: prob13.exe

Task Description

An IP address acts as a mailing address for your computer. Your computer usually exists on a local network with other computers. In order to talk to computers outside of your local network, a routing device is needed. Many of you have one of these in your home that allows you access to the web and communicate with as many computers as you want.

Communications between computers (such as accessing a web page) are made up of a series of packets (small chunks of data). Each packet contains a destination IP address. A router keeps a list of networks. The router compares this list to the IP address in each packet. When it finds a match, the packet is 'routed' to that network. There is also a default route where a packet is sent when it doesn't match any of the known networks. The default route usually sends the packet to another router that will know what to do with the packet.

Your task will be to design a router.

You will need to use the Java bitwise operator '&'. This is also called a bitwise AND. It works by converting a decimal number to binary, and then comparing each value in the binary numbers one by one using standard binary logic. Then the binary number is converted back to a decimal number.

Example 1:
```
      Decimal:     5      &      3      = 1
      Binary:      0101   &      0011   = 0001
```

Example 2:
```
      Decimal:     13     &      0      = 0
      Binary:      1101   &      0000   = 0000
```

Binary AND logic
```
1 & 1 = 1
1 & 0 = 0
0 & 1 = 0
0 & 0 = 0
```

A network can be described by an IP address and a netmask. A netmask is a filter that allows you to determine if an IP is part of a network. Perform the following steps to decide which network to use:

1. Use the Java bitwise operator '&' to perform a bitwise AND between the IP address and each netmask.
   `e.g. 192.168.1.23 & 255.255.255.0 = 192.168.1.0`
2. If the result matches one of the networks, then you are done.
3. If no match is found, then the default network is selected.

Design a router that decides what network to use when an IP address is input. Use the fixed routing table below. The program will accept an IP address on the command line, and output the target network.

Sample Input/Output

```
IP> 192.168.1.63

Route to network1.

IP> 10.1.10.1

Route to network2.

IP> 18.4.181.63

Route to network3.

IP> 2.4.6.8

Route to network3.
```

Routing Table

Network	Netmask	Network Name
192.168.1.0	255.255.255.0	network1
10.1.0.0	255.255.0.0	network2
18.4.176.0	255.255.240.0	network3
default		network3

Problem 14

Anagram Finder

8 points

JAVA: program name must be prob14.java
C /C++ program name must be: prob14.exe

Task Description

An anagram is simply the rearrangement of the letters of a word or words to make another word or other words.

Write a program to look for anagram pairs from a list of supplied words. All anagrams found will come from the supplied list of words.

Input

A list of supplied words. Some of the words will have matching anagrams in the list; some will not. There will be no more than two corresponding anagrams per word list.

Output

A list of anagram word pairs.

Sample Input/Output

```
Type test_words.txt
“DANGER”,”GARDEN”,”WHO”,”HOW”,”DO”,”ODD”

Java Anagrams < test_words.txt
WHO:HOW
DANGER:GARDEN
```

Note

Do not output word pairs more than once. For example, in the above example, “GARDEN:DANGER” would be a repeat.

Problem 15

Robert the Constructor

11 points

JAVA: program name must be prob15.java
C /C++ program name must be: prob15.exe

Task Description

Robert is a certified WCE (Word Construction Engineer) who builds words by deconstructing other words and recycling the parts. Robert receives orders for new words and BTO's them (Build To Order) using words from the community word recycling center. He breaks the words into substrings and combines these substrings to form new words. He only has room in his workshop to dismantle two words at a time and he can only use one substring from each source word. These limitations mean that Robert spends a lot of time rummaging through recycled words to locate the proper substrings.

Write a program to help Robert find two words from a list so that he can build a new word.

Sample Input

Each line of the input contains an integer N followed by N words. The first word is the BTO word that Robert needs to build. The rest are the words in the box from the community recycling center. The input ends with a single zero.

```
7 PUTTER WRANGLER TERMINAL PURE INPUT FOREIGN PULMINARY
6 FETCH REFINE WATCHER FERN CHURN WHICH
9 MANUAL MENIAL ALTER MARE SALUTE HUMANS MARTIAL METAL SPIRITUALITY
0
```

Sample Output

For each BTO word requested, the program must print the BTO word followed by two words from the recycling box that can be used to form the BTO word. The words must be printed in the order in which their substrings appear in the BTO word. You may assume there is only one correct answer.

```
PUTTER INPUT TERMINAL
FETCH FERN WATCHER
MANUAL HUMANS SPIRITUALITY
```

Problem 16

Level-Up

13 points

JAVA: program name must be prob16.java
C /C++ program name must be: prob16.exe

Task Description

Welcome to the next level!

A common feature of RPGs is that players select new skills or talents for their characters as they advance. In many games these skills have dependencies, meaning that a player cannot choose a skill until all the pre-requisites have been satisfied. Players dream of the elusive "perfect build", but when the list is long, lining up all the dependencies can be a daunting task.

Given a skill database and a list of target skills, write a program to print the sequence in which skills must be acquired to achieve the desired build.

Input

The input file is divided into two sections. The first section is a database of skills. The first line of this section indicates the number of skills in the database. Each skill is represented on a single line. Skill names may contain letters and/or numbers, but no spaces. Skill names may be followed by the names of other skills upon which the first skill depends. For example, the line:

```
      InnerBeast: Intimidation
```

defines a skill named InnerBeast that depends on the skill Intimidation. Some skills have no dependencies. The second section of the file lists the build goals, one build per line. For each build, a character's name is followed by a list of desired skills. The last line of input is the word `[End]`.

```
[Skills]
Ambidexterity
Disarm: Parry
DualWield: Ambidexterity, Precision
Feint
Focus
InnerBeast: Intimidation
Intimidation
Parry
Precision: Focus
SpringAttack
Riposte: Focus, Feint
WeaponMaster: Intimidation, Precision
Whirlwind: Precision, SpringAttack
[Goals]
Fred: DualWield, WeaponMaster
LadyEmerald: Whirlwind, Riposte, InnerBeast
[End]
```

Output

For each build, the program must print the character's name with a valid sequence of all the skills required for the build on a single line. All of a skill's dependencies must be printed before the dependent skill. The program must not print skills unneeded for the build. In many cases there will be multiple correct sequences. The program should only print one sequence per build.

```
Fred: Ambidexterity Focus Precision DualWield Intimidation WeaponMaster
LadyEmerald: Focus Precision SpringAttack Feint Intimidation Whirlwind Riposte InnerBeast
```

Problem 17

Set Operations

15 points

JAVA: program name must be prob17.java
C /C++ program name must be: prob17.exe

Task Description

Write a program to evaluate expressions containing multiple set operators. You will need to implement the following basic Set Operations:

- Complement of a set
- Union of two sets
- Intersection of two sets
- Difference of two sets

Explanation of terms:

Set:
A set is a collection of distinct elements (no duplicates) in a domain.

Universal Set:
The Universal Set consists of all the possible elements in a domain. All other sets in the system are subsets of the Universal Set. They can contain zero or more elements from the Universal Set and nothing else.

Complement:
E.g.: B = !A
The Complement (B) of set (A) is the set of all elements that are in the Universal Set but not in set (A).

Union:
E.g.: C = A | B
The Union (set C) of two sets (A and B) is the set of all elements that are in either one of the two sets or both.

Intersection:
E.g.: C = A & B
The Intersection (set C) of two sets (A and B) is the set of all elements belonging to both the sets.

Difference:
E.g.: C = A - B
The Difference (set C) of two sets (A and B) is the set of elements in the first set (A), but not in the second set (B).

Assumptions:

- The sets in the system are represented by alphabets A to Z. Universal Set is represented by U.
- The Universal Set U consists of integers from 1 to 26.
- The set operators are represented by the following symbols:

```
Complement     (Unary Operator)     !      (Exclamation)
Union          (Binary Operator)    |      (Bitwise OR)
Difference     (Binary Operator)    -      (Minus)
Intersection   (Binary Operator)    &      (Bitwise And)
```

- The precedence of the set operators is given below. The operators are arranged in decreasing order of precedence from top to bottom.

```
!  Highest Precedence
|  Second highest Precedence
-  Third highest Precedence
&  Lowest Precedence
```

Input

The following are the inputs in the same order.

- An integer indicating the number of input sets (m > 0)
- (m) lines input, one each for a set. Each of these lines consists of the following:
  - Name of the set (a capitalized alphabet from A to Z)
  - The number of elements (n) in that set (n >=0)
  - (n) integers of that set, in any order
- An integer (q) indicating the number of expressions to be evaluated
- (q) lines of input. Each line is a string containing a mix of the set names in the system, and the set operators, explained earlier. Each line will contain at least one set name. No spaces in between the set names and the set operators. Maximum length of the string is 80.

Output

You will have to evaluate each of the (q) expressions and print the number of elements in the resulting set. So the output will contain (q) lines, each containing the number of elements in the result set, followed by the items in the result set ***in increasing order***. Terminate the last line with a new line character.

Sample Input

```
3
A 4 6 8 10 14
B 2 8 19
C 5 1 2 4 10 19
2
A|B
A-B&!C
```

Sample Output

```
5 6 8 10 14 19
2 6 14
```

Problem 18

Game Pathfinder

17 points

JAVA: program name must be prob18.java
/C++ program name must be: prob18.exe

Task Description

One common complaint about video games is that sometimes the MOB pathfinding makes bad decisions. You can see this occur when a MOB "takes the long way" when travelling to a destination.

For this program we'll use a simple terrain map. Terrain comes in a variety of types like G (grass), C (cliff), W (water), B (bridge), and T (trees). MOBs can only walk on grass and bridges. Each terrain letter is followed by a single digit number that represents the terrain height. MOBs cannot walk directly from terrain of one height to another height unless there is a R (ramp) terrain. MOBs may move to any of the eight adjacent squares using a move direction of N, NE, E, SE, S, SW, W, or NW. The only exception to this rule is that bridges and ramps can only be approached and exited with the cardinal directions N, E, S, and W. In this map's coordinate system (0,0) is at the bottom left corner with x increasing to the right (east) and y increasing to top (north).

Write a program to find the shortest path for a MOB.

Input

The first line of input will be two integers W and H indicating the width and height (respectively) of the map. The next H lines will each contain W terrain type/height nodes separated by spaces. The last section of the input will contain path request lines of starting points and destinations given as (x,y) coordinates. The input ends when the path request is "(-1,-1)(-1,-1)".

```
8 8
T3 T3 T3 G3 G3 W3 G3 T2
T3 G3 C3 G2 G2 B2 G2 T2
T3 G3 R2 G2 G2 W2 G2 G1
T3 G3 C3 T2 G2 W2 G2 G1
T3 G3 G3 G3 G2 W2 B2 W1
C3 R2 G3 G2 G2 G2 G2 G1
T2 G2 G2 G2 G2 G2 G2 G1
T2 T2 T2 C3 C3 T2 G1 G1
(6,5)(3,3)
(5,6)(5,2)
(1,1)(3,5)
(-1,-1)(-1,-1)
```

Output

For each path request the program must print a sequence of moves, followed by the number of moves in the sequence. The moves must obey the terrain movement rules. The path must be the shortest possible path from the start point to the destination.

```
N W W SW W W S SE E 9
W S S S SE 5
E NE NE N NW 5
```

Problem 19

Parsing Ancient Manuscripts

19 points

JAVA: program name must be prob19.java
C /C++ program name must be: prob19.exe

Task Description

"Computer, open my room dot com." Within the next few decades new software will allow people to control computers using only voice. One of the many significant problems with writing a voice control system is dividing a string of phonetic sounds into words. You will explore ways to solve this very new problem by examining the very old problem of parsing ancient manuscripts.

In modern times, paper is abundant and inexpensive, but this hasn't always been true. In ancient times, writing materials such as papyrus and parchment were difficult and time-consuming to produce. To conserve materials, ancient writers often wrote in scriptio continua, meaning that there were no gaps between words and almost no punctuation. The first step in deciphering these ancient documents is to find the divisions between words.

Write a program that can use a supplied dictionary to parse scriptio continua text in the not-so-ancient language of English. You can consider how to apply your algorithm to the problem of voice control in your spare time.

Input

The input file will contain two sections: the dictionary of valid words and the manuscript to be parsed. The section names appear in square brackets. Multiple dictionary words may be listed in lines up to eighty characters in length. Lines in the manuscript section will have no more than eighty letters each. The manuscript input ends with a single period.

```
[DICTIONARY]
A AN AMERICA AMOR AND BLESS BLESSING BLESSINGS COMMON CONSTITUTION
DEFENCE DO DOMESTIC DOOR ESTABLISH FARE FOR FORM FORT GENE GENERAL IN
INSURE INSURED I IT JUST JUSTICE LIBERTY ME MER MORE NOR OF ON ONE
ORDAIN ORDER OUR OURSELVES PEOPLE PERFECT POST POSTER POSTERITY PROMOTE
PROVE PROVIDE SECURE SO SOFA STATE STATES THE THIS TO TOO TRANQUIL
TRANQUILITY UNION UNITE UNITED WE WELFARE
[MANUSCRIPT]
WETHEPEOPLEOFTHEUNITEDSTATESINORDERTOFORMAMOREPERFECTUNIONESTABLISHJUSTICEINSUR
EDOMESTICTRANQUILITYPROVIDEFORTHECOMMONDEFENCEPROMOTETHEGENERALWELFAREANDSECURE
THEBLESSINGSOFLIBERTYTOOURSELVESANDOURPOSTERITYDOORDAINANDESTABLISHTHISCONSTITU
TIONFORTHEUNITEDSTATESOFAMERICA.
```

Output

The program must display the manuscript text with spaces between words, with word wrapping applied at eighty characters. All the words in the text must appear in the dictionary; any division that does not identify dictionary words is invalid. There will be only one valid solution. For example, the dictionary will not contain the words JUSTICE, JUST, and ICE.
That's another problem you can solve in your spare time.

```
WE THE PEOPLE OF THE UNITED STATES IN ORDER TO FORM A MORE PERFECT UNION
ESTABLISH JUSTICE INSURE DOMESTIC TRANQUILITY PROVIDE FOR THE COMMON DEFENCE
PROMOTE THE GENERAL WELFARE AND SECURE THE BLESSINGS OF LIBERTY TO OURSELVES
AND OUR POSTERITY DO ORDAIN AND ESTABLISH THIS CONSTITUTION FOR THE UNITED
STATES OF AMERICA
```

Problem 20

Three-Letter Crossword

21 points

JAVA: program name must be prob20.java
C /C++ program name must be: prob20.exe

Task Description

A crossword is a word puzzle set on a grid of black and white squares. Each white square takes a single letter so that the letters in consecutive squares form words. The black squares provide space between words.

Write a program to fill in a crossword puzzle with a set of three-letter words.

Input

The input consists of two sections. The first section contains an integer N followed by N three-letter words. The next section contains two numbers that indicate the width and height (in that order) of the grid. That last lines contain the puzzle grid. Black spaces are indicated by '#' characters, and white spaces are indicated by '-' characters..

```
14
ASK BEN ELK GEM GOB MEN PEW SKI SKY SPA TOW YUK ZIP ZIT
11 3
- - - # - - - # - - -
- # - - - # - - - # -
- - - # - - - # - - -
```

Output

The program must fit each of the words exactly once into the grid and print the result. There may be more than one possible solution. If you try a simple brute-force search your program may take a very long time to find a solution. The judges are busy, impatient, and unlikely to wait for very long, so look for a way to optimize your search.

```
G E M # S P A # Z I T
O # E L K # S K I # O
B E N # Y U K # P E W
```

2009 HP Code Wars
EVENT EVALUATION

We need your input to evaluate our efforts, and, therefore, ask that you complete the following questionnaire.

Please rate the following	**1 Great**	**2 Pretty Good**	**3 Okay**	**4 Fair**	**5 Poor**
Overall competition					
Parking					
Registration process					
Speakers					
Competition Instructions					
Competition Room (set up, comfort)					
Time allotted					
Give-aways-Event Bags					
Door Prizes					
Lunch					
Judging					
Divisional problems					

What was your favorite part of this event?

____

What would you change about the event?

____

Other comments

____

____

____

I am a Student _____ Teacher _____, Representing ______________________ High School

Email:**____**

2009 HP Code Wars
EVENT EVALUATION

We need your input to evaluate our efforts, and, therefore, ask that you complete the following questionnaire.

Please rate the following	**1 Great**	**2 Pretty Good**	**3 Okay**	**4 Fair**	**5 Poor**
Overall competition					
Parking					
Registration process					
Speakers					
Competition Instructions					
Competition Room (set up, comfort)					
Time allotted					
Give-aways-Event Bags					
Door Prizes					
Lunch					
Judging					
Divisional problems					

What was your favorite part of this event?

____

What would you change about the event?

____

Other comments

____

____

____

I am a Student _____ Teacher _____, Representing ______________________ High School

Email: ____

2009 HP Code Wars
EVENT EVALUATION

We need your input to evaluate our efforts, and, therefore, ask that you complete the following questionnaire.

Please rate the following	**1 Great**	**2 Pretty Good**	**3 Okay**	**4 Fair**	**5 Poor**
Overall competition					
Parking					
Registration process					
Speakers					
Competition Instructions					
Competition Room (set up, comfort)					
Time allotted					
Give-aways-Event Bags					
Door Prizes					
Lunch					
Judging					
Divisional problems					

What was your favorite part of this event?

____

What would you change about the event?

____

Other comments

____

____

____

I am a Student _____ Teacher _____, Representing ______________________ High School

Email:**____**