problem 1

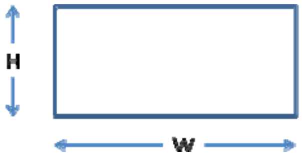
Secure the Perimeter

2 points

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

Introduction

A home owner wants to build a low fence around a rectangular part of a garden. The home owner is considering several different designs, and each design includes a rectangle of different dimensions. The cost of materials and time required to install the fence depend on the total length of fencing. Recall that the perimeter P of a rectangle of height H and width W can be calculated with this formula:



$$P = 2 \times H + 2 \times W$$

Write a program to calculate the perimeter of a rectangle.

Sample Input

The input consists of two integers: the height and width of a rectangle.

Example 1:

```
4 6
```

Example 2:

```
8 5
```

Sample Output

The program must print the perimeter of a rectangle defined by the input values and then exit.

Example 1:

```
20
```

Example 1:

```
26
```

problem **2**

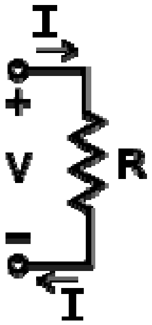
My Dear Friend VIR

3 points

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

Introduction

The most basic law used in circuit analysis is **Ohm's Law**. This law states that the current ***I*** through a conductor between two points is directly proportional to the voltage ***V*** across the two points, and inversely proportional to the resistance ***R*** between them. This relationship is expressed with the following formulas:



$$V = IR \quad \text{or} \quad I = \frac{V}{R} \quad \text{or} \quad R = \frac{V}{I}$$

The most common units of measure are ***Volts*** for voltage, ***Ohms*** for resistance, and ***Amps*** for current. However, for many small electronics projects the current is often less than 1 amp so current is often described in ***milliamps***, where 1,000 milliamps equals 1 amp.

Write a program to compute the current in milliamps through a resistor when a voltage is applied.

Sample Input

The input will consist of two floating-point (real) numbers representing a simple circuit. The first number represents a voltage value and the second represent the ohm value of the resistor. Several examples are provided below.

Example 1: `1.45 150`

Example 2: `1.17 375`

Example 3: `3.49 67`

Sample Output

The program must print the current of the circuit in milliamps.

Example 1: `9.6666667`

Example 2: `3.12`

Example 3: `52.08955`

problem **3**

Happy Numbers

4 points

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

Introduction

Most humans think of numbers as cold and unfeeling. As a matter of fact, some numbers are happy, while others are unhappy. The procedure for learning whether a number is happy or unhappy is quite simple. Starting with any positive integer, replace the number with the sum of the squares of its digits. Repeat the process until the number equals one or loops endlessly in a cycle. Numbers that converge to one are happy numbers, while those that do not are unhappy numbers. For this program loop cycles will have less than a hundred entries.

For example, starting with the number 32, the sum of squares is 13 because 3x3 + 2x2 = 13. The sum of squares for 13 is 10 because 1x1 + 3x3 = 10. The sum of squares for 10 is 1, so 32 is a happy number.

Sample Input

The input will be a single positive integer. Several examples are given here.

```
32

4565

42

86

5555
```

Sample Output

The program must print a sentence indicating if the input number is happy or unhappy, then exit.

```
32 is a happy number

4565 is an unhappy number

42 is an unhappy number

86 is a happy number

5555 is a happy number
```

problem 4

Stellar Classification

5 points

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

Introduction

When ordinary people use the word "color" to describe a star, they mean "what is the tint perceived by the eye?" One star might have a color of "pale orange", another "bluish-white". But astronomers often use the word "color" to refer to a star's ***Spectral Class***. The basic spectral classes are O, B, A, F, G, K, and M, where class O is the bluest and class M is the reddest. The spectral class can be determined from the ***B-V Color Index***, which is the difference between B and V magnitudes as measured in the UBV system:

$$\text{B-V Color Index} = m_B - m_V$$

Values for m_B and m_V are measured using passband filters. The ***B band*** covers a range of blue frequencies and the ***V band*** covers a range of green and yellow frequencies.

Write a program to classify star colors from raw B and V magnitude values.

Sample Input

Each line of input contains the name of a star, followed by two floating point values: m_B and m_V, respectively. The input ends with the word END.

```
VEGA 0.00 0.00
PROCYON 0.80 0.38
SOL 5.48 4.83
ANTARES 2.96 1.09
POLLUX 2.14 1.14
CAPELLA 0.88 0.08
END
```

Spectral Class	***B-V min value***	***B-V max value***
O	-0.350	-0.251
B	-0.250	-0.001
A	0.000	0.249
F	0.250	0.499
G	0.500	0.999
K	1.000	1.499
M	1.500	2.000

Sample Output

For each star the program must print the name of the star, the B-V color index, and the corresponding spectral class.

```
VEGA 0.00 A
PROCYON 0.42 F
SOL 0.65 G
ANTARES 1.87 M
POLLUX 1.00 K
CAPELLA 0.80 G
```

problem **5**

Greek Numerals

5 points

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

Introduction

The ancient Greeks used the letters of their alphabet to represent numbers. Decoding these numbers is an essential part of translating many ancient documents. The system shown in the chart on the right, known as Ionian numerals, used twenty-seven letters to represent the numbers 1-9, 10-90, and 100-900. Two-digit and three-digit numbers were recorded by writing the letters side by side. For example, the number twenty-seven is written KZ, a combination of the symbol for twenty, K, and the symbol for seven, Z.

Write a program to convert Greek numerals to decimal numbers.

Number Value	Greek Letter	Text Symbol
1	Α	A
2	Β	B
3	Γ	G
4	Δ	D
5	Ε	E
6	Ϝ	#
7	Ζ	Z
8	Η	Y
9	Θ	H
10	Ι	I
20	Κ	K
30	Λ	L
40	Μ	M
50	Ν	N
60	Ξ	X
70	Ο	O
80	Π	P
90	Ϙ	Q
100	Ρ	R
200	Σ	S
300	Τ	T
400	Υ	U
500	Φ	F
600	Χ	C
700	Ψ	$
800	Ω	W
900	Ϡ	3

Sample Input

Each line of input contains a sequence of text symbols that represent the Greek letters shown in the chart. The input ends with a single period character.

```
KZ
MB
3#
UQE
$L
.
```

Sample Output

The program must print the decimal number equivalent for each Greek numeral on a separate line.

```
27
42
906
495
730
```

problem **6**

Product Review Site

6 points

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

Introduction

A review web site allows users to rate products on a scale of one to five. These ratings are stored and displayed whenever other users search for the products. Ratings are displayed with a histogram showing how many of each rating the product has received and an average of all ratings. Write a program to display a ratings histogram and average from a set of ratings.

Sample Input

Each line of input is a single rating value from one to five. The input ends with the number zero.

```
4
5
3
4
2
4
3
5
2
1
3
2
4
3
2
4
3
3
2
4
3
3
5
3
1
4
3
2
4
5
4
3
4
3
3
4
3
0
```

Sample Output

The program must print a histogram that shows the number of occurrences of each rating value. The histogram must match the format shown below, where each line includes the rating value, the count in parentheses, the aligned vertical bar, and the number of ‘=’ characters based on the count. The program must also print the average rating.

```
5 ( 4) |====
4 (11) |===========
3 (14) |==============
2 ( 6) |======
1 ( 2) |==
Average rating: 3.243243
```

problem 7

Minelayer

6 points

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

hp code wars 2010

Introduction

One of the most frequently used office anti-productivity applications (you and I would call them “games”) is an application called Minesweeper. You’re helping out in the creation of a free-ware version of the game, and are tasked with setting up the playfield (a.k.a. minefield).

The playfield setup begins with mines randomly placed in a grid. Each square not containing a mine must be populated with a digit (1-9) representing the number of mines adjacent to it. Squares with no adjacent mines are filled with a period.

Sample Input

The input consists of a grid (30 columns by 15 rows of text), each square populated with either a mine “*” (asterisk) or a blank “.” (period).

```
....................*......**.
...**..*...*..................
................**.......*....
.........**...*...............
............*.......*......*..
......*....................*..
.....................*.....*..
............**.............*..
*.****................*.....*.
...............*.*.**.........
...........................*..
.......*..**.........*...**...
............*........***.*....
...*.*....*..........*.***....
...........**........***.....*
```

Sample Output

The program must output a playfield, ready for a game of Minesweeper!

```
..1221111.111......1*1....1**1
..1**11*1.1*1..1221111..112221
..12211122321112**1.....1*1...
........1**212*2221111..11211.
.....1111222*211...1*1....2*2.
.....1*1...111.....1221...3*3.
.....111...1221.....1*1...3*3.
1223321....1**1.....1221..2*31
*2****1....12221212222*1..12*1
1223321.......1*2*2**211..1221
......1111221.112122321.123*1.
......1*11**21......2*423**21.
..1121211234*1......3***6*41..
..1*2*1..1*432......3*8***2.11
..11211..12**1......2***421.1*
```

problem **8**

St. Ives

7 points

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

Introduction

A traditional English riddle goes something like this:

As I was going to St. Ives
I met a man with seven wives
Each wife had seven sacks
Each sack had seven cats
Each cat had seven kits
Kits, cats, sacks, wives
How many were going to St. Ives?

The answer, of course, is one. Solving a riddle like this requires AI techniques beyond the scope of this contest. Instead, let's leave the riddle solving to the humans (for now) and write a program that can read text formatted like the St. Ives riddle and do some basic math with it.

Sample Input

Program input is five lines with the following format:

```
A MAN WITH NUMBER COMPANIONS
EACH COMPANION HAD NUMBER CONTAINERS
EACH CONTAINER HAD NUMBER OBJECTS
EACH OBJECT HAD NUMBER ITEMS
HOW MANY SOMETHINGS?
```

Valid numbers are two through thirteen, inclusive. The numbers may or may not be equal. Singular and plural will always be different by the final -S only, even if it contradicts English spelling rules (e.g., WIFES instead of WIVES). The "something" may be any companion, container, object, or item. For example:

```
A MAN HAD FIVE EMPLOYEES
EACH EMPLOYEE HAD NINE PROJECTS
EACH PROJECT HAD SEVEN DEADLINES
EACH DEADLINE HAD THIRTEEN IMPEDIMENTS
HOW MANY DEADLINES?
```

Sample Output

The program must print the number of "somethings" followed by the "something" noun in plural form.

```
315 DEADLINES
```

problem 9

Letter Scramble

8 points

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

Introduction

In a popular board game, players lay letter tiles on a grid to form words. Points are awarded for each letter in the word, where uncommonly used letters have a higher point value than commonly used letters. Additionally, certain squares in the grid multiply the value of letters or entire words played on them. There are four modifiers: double letter score, triple letter score, double word score, and triple word score. If a word is laid across letter and word modifiers, then the letter modifiers are applied first, then the word modifiers.

Write a program to calculate point values for words placed on the game grid.

Sample Input

The input will be divided into two sections. The first section is a grid where a square is represented by two characters separated by spaces. The size of the grid is given by a single integer preceding the grid itself. Normal squares are represented with [] characters. Modifier squares begin with a 2 or 3, followed by a W for word or L for letter. The next section provides a list of words to be placed onto the grid, starting with the number of words in the section. Each word is followed by the row and column of the first letter, then by a letter H or L to indicate if the word should be placed horizontally or vertically.

```
8
[] [] [] [] [] [] [] []
[] 3W [] 2L [] [] [] []
[] [] [] [] 3L [] 2W []
[] [] [] [] [] 2L [] []
[] [] 2W [] [] [] [] []
[] [] [] 3L [] 3W [] []
[] 2L [] [] [] [] [] []
[] [] [] [] [] [] [] []
3
CRAFT 2 2 H
ZOO 6 4 V
QUARK 4 6 V
```

A - 1	N - 1
B - 3	O - 1
C - 3	P - 3
D - 2	Q - 10
E - 1	R - 1
F - 4	S - 1
G - 2	T - 1
H - 4	U - 1
I - 1	V - 4
J - 8	W - 4
K - 5	X - 8
L - 1	Y - 4
M - 3	Z - 10

Point Values

Sample Output

The program must print each word followed by the point value for that word. The point value for a word must be calculated using the rules given above. The program should treat each word individually and not be bothered if the words happen to overlap, whether correctly or incorrectly.

```
CRAFT 33
ZOO 32
QUARK 84
```

problem **10**

List Maker

10 points

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

Introduction

A common pattern in many programming tasks is to maintain a constantly changing list of items, whether those items might be characters in a text document, university applicants pending approval, or I/O requests to a storage device. Lists may support different operations, such as insertion and deletion, depending on the context in which they are used. Write a program that supports list operations ADD, INSERT, and REMOVE with the following definitions:

- ADD X – puts item X at the end of the list. This must work correctly on an empty list.
- INSERT X N – puts item X into the list just before item N.
- REMOVE X – removes item X from the list.

Sample Input

The input consists of a series of list operations. New items will always be unique in the list. The input ends with an operation named SHOW.

```
ADD NEVER
ADD COLLAR
INSERT CAT COLLAR
ADD DOG
ADD SCARES
INSERT ANYTHING CAT
REMOVE CAT
INSERT THAT SCARES
REMOVE COLLAR
INSERT WEAR ANYTHING
REMOVE DOG
ADD CAT
INSERT YOUR CAT
SHOW
```

Sample Output

The program must print the final list in the correct order.

```
NEVER WEAR ANYTHING THAT SCARES YOUR CAT
```

problem **11**

Museum Area

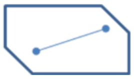
11 points

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

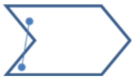
Introduction

The curator of a large museum wants to coat the museum's hardwood floors to protect the wood from damage. The curator must determine the square feet to be covered in order to estimate material and labor costs. Each room and hallway has been divided into a set of *convex* polygons and the curator has a list of the x-y coordinates of the corners of each polygon. The curator has hired you to write a program to calculate the area of a convex polygon given the x-y coordinates of its vertices (corners).

Definition: A polygon is *convex* if, for every pair of points within the polygon, every point on a straight line segment that connects the points is also within the polygon.



convex



not convex

For solving this problem you may wish to use Heron's formula for computing the area of a triangle. If you know the lengths of the three sides *a*, *b*, and *c*, then the area is:

$$A = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{where} \quad p = \frac{a+b+c}{2}$$

And, of course, the distance between two points (x1,y1) and (x2,y2) is given by the following formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Sample Input

The input consists of a series of x-y coordinates indicating the corners of a convex polygon describing a room or hallway in clockwise order. The first line of the input specifies the number of corners.

```
4
-11 -10
-11  10
 11  10
 11 -10
```

Sample Output

The program must print the area of the space enclosed by the series of points.

```
440
```

problem **12**

Abbreviation Expansion

12 points

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

Introduction

Bert has an idea for a Web 2.0 startup called ***Jitter***, which allows users to make short blog entries by typing abbreviations for common words. When generating the web pages for display the software will expand the abbreviations according to the site's lexicon. Bert believes that the time people save by typing fewer letters will encourage them to switch to his site.

Write a program to expand abbreviations into words from a lexicon.

Sample Input

The input is divided into two sections. The first section is the lexicon itself, and the second section is a ***Jitter*** user's blog entry that needs to be expanded. The sections are divided by a single | character.

```
cream chocolate every ever does do ice is fried friend friends lick
like floor favor flavor flower best but probably poorly say says that
what white our you your strawberry storyboard the | wht flvr ic crm ds
yr bst fnd lke? ur frds lk stbry, bt choc s prly th bs flr vr!
```

Sample Output

The program must print the blog entry with all the abbreviations expanded into words. A substituted word must be the shortest unique word that can be formed by adding zero or more letters (or punctuation symbols) to the abbreviation. If two or more words can be formed by adding the same number of letters, then the abbreviation should be printed as-is.

```
what flavor ice cream does your best friend like? our friends lk strawberry, but
chocolate is poorly the best floor ever!
```

problem **13**

Tetra Square

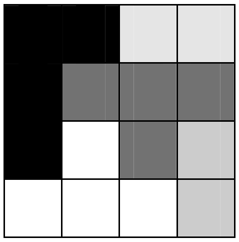
14 points

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

Introduction

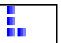
In a popular video game the player tries to fit multi-segment shapes together so that there are no gaps. In the game the player must rotate and place the shapes within a time limit, clearing full rows of segments to make space for new shapes.

Write a program to play a variant of this popular game by fitting multi-segment shapes into a four by four grid.



Sample Input

The input consists of a sequence of codes that represent the multi-segment shapes that need to be fitted together. Each code indicates not only the shape but also the orientation of the shape. That is, your program will not be required to (and should not) rotate the shapes. The list of valid codes is given in the table to the right. The input begins with a single integer indicating the number of shapes. A shape will not be repeated in the input. Two examples are shown:

J1	L1 
J2	L2
J3	L3
J4	L4
T1	Z1
T2	Z2
T3	S1
T4	S2
I1	I2

Example 1: `5 T1 T3 J3 I1 I2`

Example2: `4 L1 T1 Z1 T2`

Sample Output

The program must print a four by four grid solution where each input shape is represented by an upper-case letter. The program must assign letters to shapes in the order they appear in the input, i.e., A to the first shape, B to the second, etc. In the first example, T1 is labeled A, T3 is labeled B, J3 is labeled C, I1 is labeled D, and I2 is labeled E.

Example 1:

```
CCEE
CAAA
CBAD
BBBD
```

Example 2:

```
BBBD
ABDD
ACCD
AACC
```

problem **14**

Trace Route

16 points

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

Introduction

We explain a hypothetical computer network here, in which, there are 2 or more computers, which are connected to each other through cables. These connections can carry messages in both directions i.e., if computer A is connected to computer B through a cable then messages can be passed from A to B and B to A. For a message to be passed from one computer to the other, it takes a certain time .t. which varies from connection to connection. A given message never passes through the same connection twice.

You have to write a program, which calculates the minimum and the maximum time that a message takes to reach from a given computer A to computer B.

Input specification

1. The first line will contain an integer 'n' indicating the number of connections that exist in the network.
2. This is followed by 'n' lines of input, each describing a network connection as follows:
   <From_computer> <To_computer> <time_taken_to_pass_message>

   - The names of the computers will be an uppercase English alphabet
   - The time taken will be integer (>0 and <=10000)

3. The last line will contain two upper case alphabets, representing the two computers between which the path has to be calculated.

Output specification

The output should be 1 integer denoting the minimum cost for a message to be passed from the first computer to the second computer.

If there is no path from the first computer to the second then print NoRoute terminated by new line to denote that a route does not exist from the first computer to the second computer.

Example:

Sample Input	Sample Output
5 A B 7 B C 6 A C 5 B D 2 C D 3 A D	8

Sample Input	Sample Output
2 A B 5 C D 6 A D	NoRoute

problem 15

Asteroid Mining Robot

17 points

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

Introduction

In the future robots will travel to distant asteroids and mine ore that will be used to build more robots. These robots will have a limited supply of fuel used for excavating ore and returning home.

Sample Input

The first line of the input will contain three integers: the fuel at the beginning of the mission, the fuel required for the return trip, and the size of the robot's cargo container in kilograms. After the first line, each line represents a single excavation of ore. These lines indicate the fuel required to excavate the ore, the type of ore, and mass of the ore in kilograms. The input ends with “0 X 0”.

```
1146 529 5000
58 Zn 205
49 Fe 638
66 H2O 595
29 Pt 23
79 Si 970
26 Ag 11
15 Au 8
65 Fe 861
56 Si 872
80 Fe 1052
47 H2O 343
64 Mg 591
24 Pt 32
44 Si 662
0 X 0
```

Ore	Value per kg
Cu	1.730
Zn	1.130
H2O	3.720
Fe	0.410
Si	0.320
Mg	3.460
C	2.750
Pt	25,000.000
Au	12,260.000
Ag	190.629

Table of Ore Values

The program must optimize the cargo it transports home for maximum value. Different types of ore have different values as shown in the table above. Each line of excavation must be treated as a single unit (ore in a line cannot be split). Ore must be excavated in the order in given; for example, the robot cannot excavate line 7 until after excavating line 6. The robot excavates ore until it runs low on fuel, then it fills its cargo hold and returns home. There may be more lines in the input than can be excavated with the fuel in the robot's fuel tank. The program must ensure the robot does not excavate too much ore or there won't be enough fuel for the return trip. If the robot doesn't have enough fuel to excavate a line, then it gains none of the ore for that line.

Sample Output

The program must print and label the total kilograms and value of the cargo returned.

```
kg=4940 value=680272
```

problem 16

Ken-igma

21 points

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

hp codewars 2010

Introduction

A Ken-igma is a numeric puzzle played in a square grid that ranges in size from 3x3 to 9x9. For a grid of size NxN, only the digits 1 through N are used. For example, a 3x3 grid uses the digits 1, 2, and 3 only. Each digit may appear only once in each row and column. Additionally, the grid is divided into sections called cages, where each cage has a goal number and a math operation (for example “2 x” or “7 +”). When the operation is applied to the digits in the cells of a cage, the operation result must match the goal number. Valid operations are addition (+), subtraction (-), multiplication (x), and division (/). Cages for addition have one or more cells. Cages for multiplication have two or more cells. Cages for subtraction and division have exactly two cells and the order of the operation is not dictated by cell arrangement (i.e., “2 -” could match either “3 1” or “1 3”). Quotients must not be rounded (i.e., 4/3 does not equal 1).

Write a program to solve a Ken-igma.

Sample Input

The first line of input specifies the grid size N from 3 to 9 and the number of cages. Each line thereafter represents a single cage. The first part of the cage specifies the goal number and the operation, followed by the number of cells in the cage and then the indexes of each of the cells. Cells are indexed 1 to M (where M=N2), left to right, top to bottom (see the circled numbers in figure 1). The example input shown below represents the puzzle in figure 1.

```
3 4
2 x 2 1 4
3 / 2 2 3
7 + 3 5 6 9
2 - 2 7 8
```

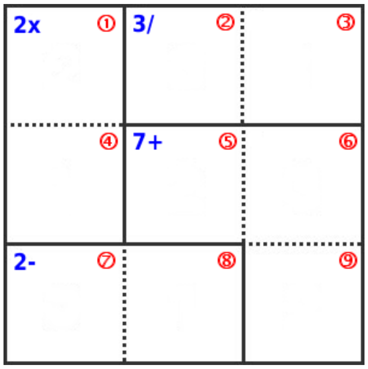


Figure 1. A 3x3 puzzle

Sample Output

The program must print the final grid of numbers that satisfy all the rules of the puzzle. There will only be one solution. The grid must be printed with a single space between each digit as shown below.

```
2 3 1
1 2 3
3 1 2
```