acm High School Programming Contest @  sponsored by: ebay

2014 University of Virginia High School Programming Contest

Welcome to the 2014 University of Virginia High School Programming Contest. Before you start the contest, please be aware of the following notes:

Rules

1. There are ten (10) problems in this packet, using letters A-J. These problems are *loosely* sorted by difficulty. As a team's solution is judged correct, the team will be awarded a balloon. The balloon colors are as follows:

Problem	Problem Name	Baloon Color
A	!terces poT	Orange
B	Tons of Bunnies, no Fibbin'	Dark Blue
C	Report Card Time	Red
D	A Serious Reading Problem	Light Green
E	Safe Zone	Yellow
F	Spawn of Ungoliant	Gold
G	The Big Eye in the Sky	Pink
H	Gimli's Gullet	Silver
I	Tools of the Trade	Light Blue
J	The Magic Word	Light Purple

2. Solutions for problems submitted for judging are called runs. Each run will be judged.

   The judges will respond to your submission with one of the following responses. In the event that more than one response is applicable, the judges may respond with any of the applicable responses.

Response	Explanation
Yes	Your submission has been judged correct.
No - Wrong Answer	Your submission generated output that is not correct or is incomplete.
No - Output Format Error	Your submission's output is not in the correct format or is misspelled.
No - Excessive Output	Your submission generated output in addition to or instead of what is required.
No - Compilation Error	Your submission failed to compile.
No - Run-Time Error	Your submission experienced a run-time error.
No - Time Limit Exceeded	Your submission did not terminate within one minute.

3. A team's score is based on the number of problems they solve and penalty minutes, which reflect the amount of time and number of incorrect submissions made before the problem is solved. For each problem solved correctly, penalty minutes are issued equal to the time at which the problem was solved plus 20 minutes for each incorrect submission. No penalty minutes are added for problems that are never solved. Teams are ranked first by the number of problems solved and then by the fewest penalty minutes.

4. This problem set contains sample input and output for each problem. However, the judges will test your submission against several other more complex datasets, which will not be revealed until after the contest. One challenge is designing other input sets for yourself so that you may fully test your program before submitting your run. Should you receive a "wrong answer" judgment, you should consider what other datasets you could design to further evaluate your program.

5. In the event that you think a problem statement is ambiguous or incorrect, you may request a clarification. Read the problem carefully before requesting a clarification. If the judges believe that the problem statement is sufficiently clear, you will receive the response, "The problem statement is sufficient; no clarification is necessary." If you receive this response, you should read the problem description more carefully. If you still think there is an ambiguity, you will have to be more specific or descriptive of the ambiguity you have found. If the problem statement is ambiguous in specifying the correct output for a particular input, please include that input data in the clarification request.

   You may not submit clarification requests asking for the correct output for inputs that you provide. Sample inputs may be useful in explaining the nature of a perceived ambiguity, e.g., "There is no statement about the desired order of outputs. Given the input: ..., would not both this: ... and this: ... be valid outputs?".

   If a clarification is issued during the contest, it will be broadcast to all teams.

6. Runs for each particular problem will be judged in the order they are received. However, it is possible that runs for different problems may be judged out of order. For example, you may submit a run for B followed by a run for C, but receive the response for C first.

   Do not request clarifications on when a response will be returned. If you have not received a response for a run within 30 minutes of submitting it, **you may have a runner ask the site judge to determine the cause of the delay. Under no circumstances should you ever submit a clarification request about a submission for which you have not received a judgment.**

   If, due to unforeseen circumstances, judging for one or more problems begins to lag more than 30 minutes behind submissions, a clarification announcement will be issued to all teams. This announcement will include a change to the 30 minute time period that teams are expected to wait before consulting the site judge.

7. The submission of abusive programs or clarification requests to the judges will be considered grounds for immediate disqualification.

Your Programs

8. All solutions must read from standard input and write to standard output. In C this is scanf/printf, in C++ this is cin/cout, and in Java this is System.in/System.out. The judges will ignore all output sent to standard error (cerr in C++ or System.err in Java). You may wish to use standard error to output debugging information. From your workstation you may test your program with an input file by redirecting input from a file:

   ```
   program < file.in
   ```

9. All lines of program input and output should end with a newline character (`\n`, `endl`, or `println()`).

10. All input sets used by the judges will follow the input format specification found in the problem description. You do not need to test for input that violates the input format specified in the problem.

11. Unless otherwise specified, all lines of program output should be left justified, with no leading blank spaces prior to the first non-blank character on that line.

12. Unless otherwise specified, all numbers in your output should begin with a - if negative, followed immediately by 1 or more decimal digits. If it is a real number, then the decimal point should be followed by as many decimal digits as can be printed. This means that for floating point values, use standard printing techniques (cout and System.out.println). The judging will check your programs with 10^{-3} accuracy, so only consider the sample output up until that point.

    In simpler terms, neither scientific notation nor commas will be used for numbers, and you should ensure you do not round or use a set precision.

13. If a problem specifies that an input is a floating point number, the input will be presented according to the rules stipulated above for output of real numbers, except that decimal points and the following digits may be omitted for numbers with no fractional component. Scientific notation will not be used in input sets unless a problem statement explicitly specifies it.

Good luck, and HAVE FUN!!!

A. !terces poT

Description

Saruman the White and his underling, Grima Wormtongue, communicate with each other using a secret code, a language as twisted as their hearts. However, the Rangers are very clever, and they realize that reversing each message will reveal their adversary's plotting. Aragorn has tasked you with decoding an archive of their messages.

Given a message on each line, output the decoded message.

Input

A series of encrpyted messages, one per line. A line with the 3-character string "END" will indicate the end of input; this line should not be decoded, and should not generate any output.

Output

The list of decrypted messages, one per line.

Sample Input

```
!edoc doog a tahW
noitacitsufbo
erafraw enirambus detcirtsernu yraurbeF fo tsrif eht no nigeb ot dnetni eW
lla sees rodroM fo drol eht ,ssertrof sih nihtiw delaecnoC
END
```

Sample Output

```
What a good code!
obfustication
We intend to begin on the first of February unrestricted submarine warfare
Concealed within his fortress, the lord of Mordor sees all
```

B. Tons of Orcs, no Fibbin'

Description

The armies of Mordor are fearsome in both stature and numbers. How did they raise such a host in so short a time? It turns out, orcs breed very quickly. For any given year, their population equals the sum of the populations from the previous two years. For example, if there are 14 orcs in year 7 and 20 orcs in year 8, then we can calculate a total population of 34 orcs in year 9, and a total population of 54 orcs in year 10. Given the populations in two previous years, calculate the population at the nth following year.

Input

Each test case is on its own line, each of the form `a b c`; Here a and b are non-negative integers denoting the number of orcs in the previous two years, and c is the number of years in the future to calculate the population in. The end of input is marked by a line of the form "0 0 0", which should produce no output. No values will exceed the value that can be stored in an `int` variable.

Output

For each test case, output one integer on its own line describing the number of orcs in the specified year. No values will exceed the value that can be stored in an `int` variable.

Sample Input

```
10 10 1
0 39 4
14 20 1
0 0 0
```

Sample Output

```
20
195
34
```

C. Report Card Time

Description

Little hobbitses go to hobbit school in the Shire. They just finished a course, which involved tea-making, meal-eating, nap-taking, and gardening. Based on the following grading scale, assign each hobbit a letter grade based on their final numerical course grade.

- A+: 97-100
- A: 90-96
- B+: 87-89
- B: 80-86
- C+: 77-79
- C: 70-76
- D+: 67-69
- D: 60-66
- F: 0-59

Input

The input will begin with a single line containing just a whole number, n, of the number of hobbits in the class, followed by n lines in the form `a b`, where a is the hobbit's name (only alphabetical characters) and b is the hobbit's grade, given as a whole number.

Output

For each test case, print out a list of every hobbits name and letter grade, each on its own line. There should be no additional white space following test cases.

Sample Input

```
5
Bilbo 13
Sam 90
Pippin 78
Frodo 97
Merry 70
```

Sample Output

```
Bilbo F
Sam A
Pippin C+
Frodo A
Merry C
```

D. A Serious Reading Problem

Description

Gandalf's search for the history of the One Ring took him to the library of Minas Babel, under the care of Archivist Borges. This library contains every possible book - true and false, nonsensical and insightful - and Gandalf must use all of this wisdom to find the true account of the Ring.

Given that there are C possible characters, W characters in a line, L lines on a page, and P pages in a book, tell how many books are in the library of Minas Babel, if each possible book appears exactly once.

For example, if the library were to contain books with 2 characters (a,b), 3 characters per line, one line per page, and one page per book, then the following books would be in the library: aaa aab aba abb baa bab bba bbb

Input

Each test case is on its own line, each of the form `C W L P`; all are non-negative integers. The end of input is marked by a line of the form "0 0 0 0", which should produce no output.

Output

For each test case, output a new line containing a single integer, which should represent the number of books in the library.

Sample Input

```
4 2 2 2
4 3 2 2
0 0 0 0
```

Sample Output

```
65536
16777216
```

E. Safe Zone

Description

Saruman's army has mounted their seige upon Helm's Deep, and are unleashing volley after volley of arrows on its walls – but the Rohirrim have only enough shields to defend a segment of Helm's Deep's walls! The Uruk-hai orcs are beating at the gate, and Saruman himself assaults your walls with sorcerous missiles cast from his tower at Isengard. You are a soldier in King Theoden's army – with the walls shaking under your very feet, you must make your way to cover before the next volley of arrows land!

From left to right, the segments of the wall are labelled 0 to (N-1), and you are currently at segment J. The Rohirrim shields cover from P to Q, and you have time to make exactly K steps (each step can be either left or right) before the next volley of arrows lands (the wall's rumbling makes it impossible to stand still). How many different combinations of steps can get you to safety?

Input

Input is given in the form of a single line per test case, with the order `N P Q J K`. The size of the wall is N, the safe zone is from P to Q, the starting position is at J, and the number of steps is K. A line with N equal to 0 marks the end of input and should not produce output.

Note that you cannot walk off the end of the wall. This means that if you are at either end of the wall, you must walk toward the center.

Output

For each test case, output on one line the number of ways to end up in the safe zone in K steps.

Sample Input

```
4 0 1 2 2
4 0 1 2 3
16 2 3 7 13
0 0 0 0 0
```

Sample Output

```
1
3
715
```

F. Spawn of Ungoliant

Description

The spider children of Ungoliant are spreading through Mirkwood. Each of the giant spiders cannot move from the tree which it inhabits; however, each can lay eggs in the trees adjacent to it. Each egg will rapidly hatch and grow into another spider, who will then lay their own eggs.

Given a map of Mirkwood, indicating where spider-filled and spider-free trees are, determine how far the spider infestation will spread.

For this problem, adjacency is one of the four cardinal directions (north, south, east, and west); two trees connected via a diagonal are not considered adjacent.

Input

The first line of an input case is of the form `W H`, where W is the width of the map and H is the height. The next H lines contain strings of length W specifying the layout of Mirkwood. An S character represents a spider-infested tree, and an T character represents a spider-free tree. A . (period) signifies open space on the forest floor, where there is no tree.

The end of input is signified by two zero's, which is a case that should not be processed.

Output

For each test case, print out the map of Mirkwood once the spiders have spread as far as possible. There should be no blank lines between maps.

Sample Input

```
3 4
T..
TST
..T
TTT
5 5
T.T.T
.T.T.
..S..
.T.T.
T.T.T
0 0
```

Sample Output

```
S..
SSS
..S
SSS
T.T.T
```

```
.T.T.
..S..
.T.T.
T.T.T
```

G. The Big Eye in the Sky

Description

The Eye of Sauron is in the middle of Mordor. He needs to track the tricksy hobbitses as they trek across the land; to do so, the Eye at the top of the Tower must rotate so that it has a good view.

The eye is at coordinates (0,0) facing due east. Given a coordinate in the first quadrant, output the number of degrees from the x-axis to rotate the Eye so that it faces the hobbits.

Input

The input is a series of lines representing integer coordinates of the hobbitses as an x and y value. All (x, y) coordinates will be in the first quadrant. The coordinate (0,0) marks the end of input.

Output

For each test case, output to a newline the number of degrees (rounded to the closest integer) to rotate the Eye. Note that each test case is rotating from the x-axis, not from the previous orientation of the Eye.

Sample Input

```
5 5
9 15
12 6
0 4
0 0
```

Sample Output

```
45
59
27
90
```

H. Gimli's Gullet

Description

Legolas and Gimli are packing rations before setting off from the Woodland Realm, but the kitchen seems to have underestimated the belly of a dwarf! Gimli absolutely refuses to set off until the matter is settled and his pack is at its maximal hunger-satisfying potential, lest his Dwarven appetite go unsated.

Input

Input consists of several test cases, each over multiple lines. The first line of every test case contains a single non-negative integer, denoting the space (in cubic centimeters) remaining in Gimli's pack. The second line contains a single non-negative integer, M, describing the number of food items available. The next M lines each consist of two non-negative integers, describing the volume (also in cubic centimeters) and caloric content of a food, respectively. These lines are sorted in increasing order of volume.

A test case starting with 0 cubic centimeters denotes end of input, and should not produce output.

Output

For each test case, output a single space-separated line with M entries. Each entry in the line is the quantity that Gimli should order of the corresponding food from the kitchen in order to fill his pack with the maximum amount of food.

Sample Input

```
6000
4
230 100
350 500
480 900
1400 2000
760
2
10 180
200 300
0
```

Sample Output

```
0 2 11 0
76 0
```

I. Tools of the Trade

Description

King Theoden has responded to the raids upon his lands and the threat of Saruman by withdrawing his people into Helm's Deep. Here they hope to make their stand against the forces of evil, but they've run into a problem! The armory doesn't have enough weapons of each type to arm everyone with the weapon with which they are most proficient. Instead, the quartermaster must figure out how to distribute the weapons they do have so that their combat ability is maximized, and so that they can repulse the Uruk-hai armies.

Input

The first line of the input is an integer giving the number of test cases. Each test case begins with a line of the form "N M", where N is the number of weapon types and M is the number of soldiers. N lines follow, each of the form C T, where C is the character representing the weapon, and T is the number of that weapon available. M lines follow that, describing the proficiencies of each soldier (in decreasing order) in terms of the character representation of the weapons.

Output

For each test case, give an integer on its own line as output. This number is the total inexperience for the best (minimal) assignment of weapons. This is measured as the total of the number of steps from their favored weapon that each soldier is assigned; for example, a soldier with the preferences ABCD assigned weapon C would increase the total inexperience by 2, whereas if assigned B he would increase the total inexperience by 1.

Sample Input

```
2
4 6
A 1
B 2
C 3
D 4
ABCD
BDCA
BDCA
BACD
CDBA
ABCD
4 3
E 1
F 1
B 1
A 1
ABEF
ABEF
ABEF
```

Sample Output

```
3
3
```

J. The Magic Word

Description

> Do not meddle in the affairs of wizards, for they are subtle and quick to anger.
> -Gildor

Gandalf has to work with the arcane arts regularly, to aid in his fight against Sauron. However, this can often be quite dangerous, as an arrangement of runes can have unintended consequences. He has tasked you, one of his followers, to design a system that could tell him in advance the magical outcome of a given set of runes.

The input takes the form of a series of instructions, one per line, all part of the same spell. As this is somewhat elvish magic, all of the instructions are in elvish. Gandalf has provided you with the meanings of the instructions below, so that you can better understand the magical patterns.

Power Word	Translation	Effect
arth [lbl]	region	goto [lbl]
pen [var1] [var2] [lbl]	less	if [var1] < [var2], goto [lbl]
muindor [var1] [var2] [lbl]	brother	if [var1] == [var2], goto [lbl]
sad [lbl]	place	label [lbl]
tangado [var]	to establish	declare [var] as an int, set to 0
ost [var1] [var2]	stronghold	declare [var1] as an int array of size [var2]
teithant [string]	to write	print [string] to stdout, followed by a newline
canad [var]	to shout	print [var] to stdout, followed by a newline
anno [var1] [var2]	give	var1 := var2
aderthad [var1] [var2]	to reunite	var1 := var1 + var2
adlanna [var1] [var2]	to slant	var1 := var1 / var2 (integer division)
adlegi [var1] [var2]	to release	var1 := var1 - var2
athrado [var1] [var2]	to cross	var1 := var1 * var2
awarthad	to abandon	exit

A few notes:

- [var] and [var n] may be of the following forms:
  1. An integer constant, e.g. 6; if attempting to assign to an integer constant, do nothing and go to the next command.
  2. A variable which has been declared with tangado, e.g. shadowfax; treat as an int would be in C or Java.
  3. An index in an array which has been declared with ost, e.g. moria[4]; the index of the array is itself a [var].
- [string] refers to a quote-delimited string constant, which cannot contain quotes and cannot contain newline characters.
- [lbl] refers to a label; labels may share names with [var]iables, but should be treated completely independently. labels are not pre-declared, so it may be advantageous to check the rest of the spell first to figure out label locations before interpreting it.
- The `arth` instruction means that the next command interpreted is the one immediately after the [lbl] indicated.

- The `sad` command has no effect on variables, but is used as a destination for arth commands. Note that arth commands may refer to sad commands further down in the spell.
- The `ost` command, for declaring arrays, initalizes all the cells to zero.
- The names of all variables and labels are strings consisting of only uppercase and lowercase letters.

For example, the following spell would print out "Hello World" (without the quotes) and exit.

```
teithant "Hello World"
awarthad
```

The following spell would print out "to Rohan!" (without the quotes) and exit.

```
tangado shadowfax
anno shadowfax 4
aderthad shadowfax 3
pen shadowfax 6 mordor
teithant "to Rohan!"
awarthad
sad mordor
teithant "to Mordor!"
awarthad
```

Input

The input is a series of instructions, one per line. The instructions are terminated by a single line with the string,"END".

Output

For each test case, print the output of the interpreted instructions (i.e. the teithant and canad lines), with each command putting its output on its own line.

Sample Input

```
tangado x
anno x 3
aderthad x 2
canad x
adlanna x 2
canad x
adlegi x 1
canad x
athrado x 3
canad x
tangado i
ost sequence 10
anno sequence[0] 0
anno sequence[sequence[0]] 0
```

```
anno sequence[2] 1
anno sequence[3] 1
anno sequence[4] 1
anno sequence[5] 1
anno sequence[6] 0
anno sequence[7] 3
anno sequence[8] 4
anno sequence[9] 5
sad loop
muindor i 0 end
muindor sequence[i] 0 ttthti
muindor sequence[i] 1 gar
muindor sequence[i] 2 ggar
muindor sequence[i] 3 wdys
muindor sequence[i] 4 th
muindor sequence[i] 5 ti
muindor sequence[i] 6 tmwhg
muindor sequence[i] 7 abom
muindor sequence[i] 8 sfh
muindor sequence[i] 9 lnancb
sad ttthti
teithant "They're taking the hobbits to Isengard!"
arth pool
sad gar
teithant "Gard"
arth pool
sad ggar
teithant "Ga-gard"
arth pool
sad wdys
teithant "What did you say?"
arth pool
sad th
teithant "The hobbits"
arth pool
sad ti
teithant "To Isengard!"
arth pool
sad tmwhg
teithant "Tell me, where is Gandalf, for I much desire to speak with him."
arth pool
sad abom
teithant "A balrog of Morgoth"
arth pool
sad sfh
teithant "Stupid fat hobbit!"
arth pool
sad lnancb
teithant "Leave now, and never come back!"
arth pool
sad pool
aderthad i 1
arth loop
sad end
```

```
awarthad
END
```

Sample Output

```
5
2
1
3
They’re taking the hobbits to Isengard!
They’re taking the hobbits to Isengard!
Gard
Gard
Gard
Gard
They’re taking the hobbits to Isengard!
What did you say?
The hobbits
To Isengard!
```