

A	Bailout Bonus	bonus.in	bonus.out	2 seconds	64 MiB
B	Risk Assessment	risk.in	risk.out	2 seconds	64 MiB
C	Stock Trading	trading.in	trading.out	2 seconds	64 MiB
D	Foreclosure Borough	borough.in	borough.out	2 seconds	64 MiB
E	Toxic Assets	assets.in	assets.out	2 seconds	64 MiB
F	Stimulus Package	stimulus.in	stimulus.out	10 seconds	64 MiB

# Problem A: Bailout Bonus

**File Name:** bonus.cpp|bonus.java

**Input File:** bonus.in

## Description

In order to prevent many financial companies from collapsing, the US Government (and several other governments) “bailed them out”, in the sense that they provided multi-billion dollar emergency loans. After this happened, several of the companies went ahead and paid their executives massive performance bonuses. After a large public outcry, one approach that was taken to recoup these bonuses for the government was to impose a massive 95% tax on bonuses for employees of companies that received bailout money<sup>1</sup>. Here, you are to compute how much money the government will recover.

You will be given a list of the companies that received bailout money, and the proposed taxation rate  $r$  (such as 95%), written as an integer (in this case, 95). You will also be given a list of executives with their company affiliation and proposed bonus. Each executive working for a bailed out company will have to pay an  $r$  fraction of his bonus back (rounded down to the nearest integer). For simplicity, we assume that executives working for companies that weren't bailed out have to return none of their bonus.

## Input

The first line contains the number  $K$  of data sets. This is followed by  $K$  data sets, each of the following form:

The first line contains four integers  $C, B, n, r$ .  $1 \leq C \leq 500$  is the total number of companies.  $0 \leq B \leq C$  is the number of these companies that were bailed out.  $1 \leq n \leq 10000$  is the total number of executives, and  $0 \leq r \leq 100$  is the taxation rate in percent.

This is followed by a line containing  $B$  integers, the indices (between 1 and  $C$ ) of the bailed out companies. Finally, there are  $n$  lines, each describing one executive. Each such line consists of two integers  $c_i, p_i$ .  $1 \leq c_i \leq C$  is the company that executive  $i$  is working for, and  $p_i$  is the bonus payment that executive  $i$  received.

## Output

For each data set, first output “Data Set  $x$ :" on a line by itself, where  $x$  is its number. Then, output, on a line by itself, the total amount of money recovered by the government via taxes. Each data set should be followed by a blank line.

## Sample Input/Output

Sample input bonus.in

```
2
4 2 6 95
1 4
1 1000000
3 25000
3 102425
4 567444
1 150
4 7000
2 0 2 99

1 500000
2 500000
```

Corresponding output

```
Data Set 1:
1495863

Data Set 2:
0
```

<sup>1</sup>It remains to be seen whether this will be judged constitutional.

# Problem B: Risk Assessment

File Name: risk.cpp|risk.java

Input File: risk.in

## Description

One way in which investors make informed decisions about what companies to put their money in (e.g., by buying stock) is by reading independent assessments. There are several services writing these up, and grading stocks. During the heydays of financial wheeling and dealing, apparently, most companies were judged to be sound investments, even when they had no discernible business model. All that it would take was a sentence like “a high-yield prospect”, and investors would follow like lemmings. In fact, a computer program could probably do the same. And guess what: you get to write that program.

Your program will get a piece of text about one or more companies. There are certain *quality words* with positive or negative connotations. For instance “trouble” is negative, and “promising” is positive. For each such word, we will have a basic quality score. However, sentences can also contain the modifiers “not”, “very”, “extremely” and “slightly”. If the word “not” appears in a sentence, it changes the value of *all* quality words in that sentence to 0. “Very” doubles the value of each quality word in its sentence, “extremely” triples the value of each quality word, and “slightly” halves the value. These multipliers accumulate multiplicatively: for instance, the sentence “AIG very trouble extremely very.” assigns “AIG” a score of -12 (assuming “trouble” has a score of -1). Modifiers apply only within a sentence; a sentence is defined as ending with a ‘.’ (our texts will not contain ‘,’ ‘;’, or ‘:’ or other punctuation).

A piece of text can be about multiple companies. The way we find out what quality words apply to what company is as follows: from the moment in the text at which a company name appears until the next name of a company (or the end of the text), we assume that all words refer to that company. All words before the first company name talk about no company. Notice that multiple companies can be in the same sentence. For instance “AIG trouble Pixar promising very.” assigns a score of -2 to “AIG” and +2 to “Pixar” (assuming “trouble” counts for -1 and “promising” for +1), because “very” applies to both “trouble” and “promising” (being in the same sentence with both).

If a company appears in multiple sentences (or multiple blocks), the scores from those sentences are added.

## Input

The first line contains the number  $K$  of data sets. This is followed by  $K$  data sets, each of the following form:

The first line contains three integers  $C, Q, L$ .  $1 \leq C \leq 100$  is the number of companies,  $1 \leq Q \leq 100$  the number of quality words, and  $1 \leq L \leq 1000$  the number of lines of text. This is followed by  $C$  lines, each containing the name of a company (consisting of letters and possibly hyphens). Next come  $Q$  lines. Each of these lines contains first a quality word  $w_i$  (also possibly containing hyphens), then a space, and then a floating point number  $q_i$ , the quality score of word  $i$ .

Finally, there are  $L$  lines of text. Each line has at most 80 characters. All characters are either upper or lower case characters, hyphens (which are part of words), ‘.’ or space. Company names only match if the case is correct, i.e., “AIG” does not match “aig”. However, quality words or modifiers match irrespective of case, e.g, “trouble” matches “tRoubLE” and “Very” matches “veRY”. Our input will ensure that no two quality words are the same, and no company name is the same as any other company name or quality word.

## Output

For each data set, output “Data Set  $x$ :” on a line by itself, where  $x$  is its number. Then, output the quality scores for all  $C$  companies (rounded to two decimals), one per line. They should be in the same order as the company names were given in the input. Each data set should be followed by an empty line.

## Sample Input/Output

See next page

Sample input risk.in

Corresponding output

```
1
4 5 7
AIG
Pixar
Wells-Fargo
Microsoft
risky -0.5
promising 1
bad -1.5
blue-chip 1.5
trouble -1
In these risky economic times make good
investments. Not bad ones. Microsoft is still very blue-chip stock but
AIG looks risky. Pixar is not bad up-and-coming. Could even be
slightly promising. Not very risky. Not blue-chip
though. wells-fargo is extremely bad.
Considering MICROSOFT more it is promising.
Wells-Fargo is not blue-chip in fact extremely risky. Bad. Very very bad.
```

```
Data Set 1:
-1.00
-3.00
-7.50
3.00
```

# Problem C: Stock Trading

File Name: trading.cpp|trading.java

Input File: trading.in

## Description

Stock trading is a great way to make money if you think that the high-stakes tables in Las Vegas are not risky enough for you. The idea of trading stock is quite simple: you try to buy stock when the price is low, and sell when the price is high. Only thing is: you don't know whether the price will go up or down next. But if you did, the problem would become much easier. Here, we are going to explore how to find an optimal investment strategy in retrospect.

There are  $n$  stocks total, and  $D$  days. For each day, you are given the cost of each stock  $i$ . You start out with a capital of  $C$  dollars. On each day, you can buy or sell arbitrary amounts of combinations of stock, of course not exceeding your capital. However, there is one more rule, namely, that you can make at most  $t$  trades total (where you will be given the parameter  $t$ ). A trade is the act of either buying or selling one kind of stock. We assume that you do not need to buy stock in integer amounts, e.g., you can buy 1.364 units of a stock, if you want. If you buy multiple kinds on the same day, you are using multiple trades. Your goal is to maximize the total amount of money you have after  $D$  days. (This only counts cash; any stock you still hold after  $D$  days is useless.)

## Input

The first line is the number  $K$  of input data sets, followed by the  $K$  data sets, each of the following form:

The first line contains integers  $n, D, t$  and a floating point number  $C$ .  $1 \leq n \leq 100$  is the number of stocks,  $1 \leq D \leq 100$  the number of days, and  $0 \leq t \leq 100$  the number of trades you can make.  $C \geq 0$  is your starting capital.

This is followed by  $D$  lines, each containing  $n$  floating point numbers  $r_{d,i} \geq 0$ .  $r_{d,i}$  is the price at which stock  $i$  trades on day  $d$ . All the prices for one day will be on the same line.

## Output

For each data set, output "Data Set  $x$ :" on a line by itself, where  $x$  is its number. On the next line, output the maximum amount of money you can have after the  $D$  days (rounded to two decimals), having made at most  $t$  trades total. Each data set should be followed by a blank line.

## Sample Input/Output

Sample input trading.in

```
2
1 2 1 100.0
1.0
98.0
3 8 6 100.0
1 1 9.5
2 10.5 2
4 15 4.1
4 15 5
5 5.1 1.8
6 8 1
6.5 2.5 1
6.5 2 1
```

Corresponding output

```
Data Set 1:
100.0

Data Set 2:
4265.62
```

# Problem D: Foreclosure Borough

File Name: borough.cpp|borough.java

Input File: borough.in

## Description

One of the causes (and consequences) of the financial crisis is the large number of home foreclosures. The fact that many derivatives were backed by mortgages, on which the homeowners then foreclosed, is part of the problem that financial institutions are facing. In turn, by losing work and other income sources, individuals cannot make their mortgage payments any more, and need to foreclose their homes. As you might expect, it's a bit of a vicious cycle.

Some areas are harder hit than others by this. There are areas where a significant percentage of homes is in foreclosure right now. You are to write a program that finds out which areas are most affected.

You will be given the coordinates of houses, and whether or not they are in foreclosure. Then, you will be given simple (i.e., not self-intersecting) polygons representing different boroughs or areas. These polygons might overlap. You are to compute the percentage of houses in foreclosure in each of them, and to sort the boroughs by this foreclosure rate.

## Input

The first line contains the number  $K$  of data sets. This is followed by  $K$  data sets, each of the following form:

The first line contains two integers  $H, B$ , the number of houses ( $1 \leq H \leq 1000$ ) and boroughs ( $1 \leq B \leq 100$ ), respectively. This is followed by  $H$  lines, each describing a house, by giving its  $x$  and  $y$  coordinates (floating point numbers), and a character 'Y' (in foreclosure) or 'N' (not in foreclosure).

This is followed by the description of  $B$  boroughs. Each borough is described by a line. The first entry of the line is an integer  $c \geq 3$ , the number of corner points of the polygon. This is followed by  $2c$  floating point numbers. Each pair of floating point numbers describes one of the  $c$  corners, in the order  $x_1, y_1, x_2, y_2, \dots, x_c, y_c$ . The points will be given in counter-clockwise order.

Our inputs will ensure that no house lies exactly on the border of a borough. Also, each borough will contain at least one house.

## Output

For each data set, output "Data Set  $x$ :" on a line by itself, where  $x$  is its number. Then, on  $c$  lines, output the foreclosure rates for all boroughs. Sort the lines by non-increasing foreclosure rates, breaking ties by lower borough number. Each line should first contain the number of the borough, and then the foreclosure rate, rounded to two decimals. Each data set should be followed by an empty line.

Sample input borough.in

```
1
8 3
0 0 Y
0 1 N
1 0.5 N
0.7 0.7 Y
-0.5 0 N
-5 0 N
0.7 0 Y
3.4 0.95 N
3 -1 -0.5 1 -0.5 -1 0.75
4 -0.2 -0.5 1 -0.2 3 1.5 -0.2 1.5
5 0.5 0 1.5 -0.2 4 1.2 1 0 0.5 1.5
```

Corresponding output

```
Data Set 1:
3: 66.67%
2: 60.00%
1: 50.00%
```

# Problem E: Toxic Assets

File Name: `assets.cpp|assets.java`

Input File: `assets.in`

## Description

One of the reasons that home foreclosures have had such a huge impact on the economy is that various financial institutions invented mortgage-backed derivatives. Essentially, the right to the mortgage payments was packaged as an investment, i.e., sold to other investors in small chunks. These chunks might then be mixed with other investments, and repackaged and sold further. Thus, you have all kinds of derivatives whose real worth depends on mortgages hard to trace<sup>2</sup>. If you find out how the basic “building blocks” upon which all the derivatives rely are changing in value, it might not be at all obvious to find out what your portfolio is really worth now. You might have to track it through a chain of derivatives.

Here is how we will model this. You have some basic investments, such as mortgages. For each of those, we assume that one unit was worth \$1 before, and we will tell you how much it is worth now. Then, we have derivatives. Each derivative will be composed of one or more basic investments or other derivatives. You will get a list of other investments, together with the percentage that they contribute to the derivative (those percentages will add up, of course). We will ensure that each derivative will only contain basic investments, or derivatives with a lower number. In particular, there will be no cycles. Finally, we will give you the percentages of investments in your portfolio. Your goal is to calculate how much one unit of your portfolio (which was initially worth \$1) is worth now.

## Input

The first line contains the number  $K$  of data sets. This is followed by  $K$  data sets, each of the following form:

The first line of each data set contains two integer numbers  $B, D$ , the number of basic investments ( $1 \leq B \leq 100$ ) and the number of derivatives ( $0 \leq D \leq 100$ ). We will number the basic ones  $1, \dots, B$ , and the derivatives  $B + 1, \dots, B + D$ . This is followed by a line with  $B$  non-negative floating point numbers, the current values of one unit of each of the basic investments in order.

Next will be  $D$  lines. The  $d^{\text{th}}$  line will consist of  $B + d - 1$  non-negative floating point numbers, describing the percentage of investment  $B + d$  that is made up of investments  $1, \dots, B + d - 1$ . These numbers will always add up to 1. No derivative ever contains parts of a derivative with a higher number.

Finally, there will be one more line with  $B + D$  non-negative floating point numbers (adding up to 1). This line tells you what percentage of your portfolio is made up of each of the investments.

## Output

For each data set, output “Data Set  $x$ :” on a line by itself, where  $x$  is its number. On the next line, output the current value of one unit of your portfolio, rounded to two decimals. Each data set should be followed by one empty line.

## Sample Input/Output

Sample input `assets.in`

```
1
3 4
0.01 1.1 0.4
0.5 0.25 0.25
0.9 0 0 0.1
0 0 0 0.5 0.5
0 0.5 0 0 0.3 0.2
0 0 0.2 0.2 0.2 0.2 0.2
```

Corresponding output

```
Data Set 1:
0.33
```

<sup>2</sup>This description is certainly an oversimplification. It's as best as we understand what is happening. In fact, another reason that the economy is not doing so well is that very few people actually understood how these derivatives work.

# Problem F: Stimulus Package

File Name: stimulus.cpp|stimulus.java

Input File: stimulus.in

## Description

In an attempt to mitigate the impacts of the economic crisis, the US Government has passed a massive stimulus package. The idea is that since other investors are shying away from spending money (partly due to lack of money, partly due to fear), the government should invest in worthwhile causes which will create jobs at the same time, thus helping the economy recover. Of course, this leads to a big decision of which projects are worthy. There are several variables of importance here: how many jobs are created, how much the project costs, and how much permanent infrastructure or knowledge is created. For job creation, there is also the consideration that some projects generate jobs immediately (e.g., building a bridge), while others might generate jobs in the future (e.g., computer science research). Thus, there is a complicated tradeoff. (And we haven't even talked about geographic balance yet.)

In this problem, you are to write a program to find in a sense the “best” selection of projects to fund. For each project  $i$ , you will be given its cost  $c_i$ , infrastructure gain  $g_i$ , and job creation for the next  $Y$  years  $j_{i,1}, j_{i,2}, \dots, j_{i,Y}$ . You are also given a total budget  $B$ , and job creation targets  $J_1, J_2, \dots, J_Y$ . You are to find out if there is a selection of projects that will meet all job creation requirements without exceeding the budget. If there is such a selection, you are to output the maximum total infrastructure gain of any such selection.

## Input

The first line is the number  $K$  of input data sets, followed by the  $K$  data sets, each of the following form:

The first line of each data set contains three integers  $n, Y, B$ .  $n \leq 20$  is the total number of projects you are considering,  $1 \leq Y \leq 50$  is the number of years you are looking into the future, and  $0 \leq B \leq 1000000000$  is your total budget.

This is followed by one line with  $Y$  integers, containing the job creation targets  $J_1, \dots, J_Y$ . Finally, there are  $n$  lines, each containing  $Y + 2$  integers. The  $i^{\text{th}}$  line contains first the job creation numbers  $j_{i,1}, \dots, j_{i,Y}$  for the  $i^{\text{th}}$  project, then the cost of the project, and finally the infrastructure gain  $g_i$  for project  $i$ .

## Output

For each data set, output “Data Set  $x$ :” on a line by itself, where  $x$  is its number. On the next line, output either “No selection.” (if there is no selection meeting all job creation requirements), or the best total infrastructure gain among all selections that meet the job creation requirements. Each data set should be followed by a blank line.

## Sample Input/Output

Sample input stimulus.in

Corresponding output

```
2
6 6 100
2 2 2 2 2 2
3 0 3 0 3 0 50 100
1 1 0 0 0 0 20 10
0 0 1 1 0 0 20 10
0 0 0 0 1 1 20 10
0 3 0 3 0 3 55 30
1 1 1 1 1 1 40 0
4 2 10
2 2
2 1 7 1
0 1 5 1
1 1 4 2
1 0 2 3
```

```
Data Set 1:
30

Data Set 2:
No selection.
```