Machine Learning 2020-ITMO-CT

A. Cross-validation

1 second, 256 megabytes

Split the set of N objects, each of which belongs to one of the M classes, into K parts. Each object must belong exactly to one part so that the sizes of these parts, as well as the distribution of classes in these parts, are balanced. Formally, let $cnt(x, c)$ be the number of objects with class c falling into the part x, then $\forall x, y, c : |cnt(x, c) - cnt(y, c)| \leq 1$ and $\forall x, y : \left|\sum_c cnt(x, c) - \sum_c cnt(y, c)\right| \leq 1$.

Input

First line: three integers N, M, K ($1 \leq N \leq 10^5$, $1 \leq M, K \leq N$) — the number of objects, classes and parts.

Second line: N integers C_i ($1 \leq C_i \leq M$) — class of i-th object.

Output

Print K lines. Each line x begins with an integer S — the size of the part of x. Next are the S integers — the numbers of the objects falling into the x part. Objects are numbered from one.

input

```
10 4 3
1 2 3 4 1 2 3 1 2 1
```

output

```
4 1 4 9 10
3 2 3 5
3 6 7 8
```

The first part contains four objects, two of them are of the first class, one of the second and one of the fourth. In the second and third parts, there are three objects of the first three classes.

B. F-score

1 second, 256 megabytes

As a result of an experiment on classification into K classes, a confusion matrix CM was obtained, where $CM[c, t]$ is the number of objects of class c that were classified as t. Using this confusion matrix, calculate the macro and micro F-measures weighted averaged over the classes.

Input

The first line contains an integer K — the number of classes ($1 \leq K \leq 20$). Next comes the K lines — a description of the confusion matrix. Each row of c contains K integers — the c-th row of the confusion matrix. $\forall c, t : 0 \leq CM[c, t] \leq 100$ and $\exists c, t : CM[c, t] \geq 1$.

Output

Print two real floating-point numbers — a weighted average averaged over the classes of macro and micro F-measure. The absolute error of the answer should not exceed 10^{-6}.

input

```
2
0 1
1 3
```

output

```
0.6
0.6
```

input

```
3
3 1 1
3 1 1
1 3 1
```

output

```
0.326860841
0.316666667
```

In the first example, classes are distributed as 1:4. Precision, recall, and the F-measure of the first class are 0, and the second is 0.75. Moreover, the average accuracy, completeness and F-measure are 0.6.

C. Nonparametric regression

2 seconds, 256 megabytes

Implement a nonparametric regression algorithm that supports various functions of distances, kernels, and windows. A description of the kernels can be found here: https://en.wikipedia.org/w/index.php?oldid=911077090

Input

The first line contains two integers N and M — the number of objects and attributes ($1 \leq N \leq 100$, $1 \leq M \leq 10$).

Next N lines are a description of the data set. Each line i contains $M + 1$ integers $d_{i,j}$ ($-100 \leq d_{i,j} \leq 100$) — a description of the i-th object. The first M of these numbers are attributes of the i-th object, and the last is its target value.

The next line describes the query object q. It consists of M integers $d_{q,j}$ ($-100 \leq d_{q,j} \leq 100$) — attributes of the object q.

Next are three lines consisting of lowercase Latin letters.

The first one is the name of the distance function used: *manhattan*, *euclidean*, *chebyshev*.

The second is the name of the kernel function: *uniform*, *triangular*, *epanechnikov*, *quartic*, *triweight*, *tricube*, *gaussian*, *cosine*, *logistic*, *sigmoid*.

The third is the name of the type of window used: *fixed* — a fixed-width window, *variable* — a variable-width window.

The last line contains the window parameter: the integer h ($0 \leq h \leq 100$) — the radius of the window of a fixed width, or the integer K ($1 \leq K < N$) — the number neighbors counted for a variable-width window.

Output

Print a single floating-point number — the result of the query.

input

```
3 2
0 2 1
1 1 0
2 0 1
0 0
euclidean
uniform
fixed
2
```

output

```
0.0000000000
```

input

```
3 2
0 2 1
1 1 0
2 0 1
0 0
euclidean
gaussian
variable
2
```

output

```
0.6090086848
```

D. Linear regression

1 second, 256 megabytes

Find the equations of a line approximating the position of objects from a given data set.

Input

The first line contains two integers N ($1 \le N \le 10^4$) — the number of objects in the training set and M ($1 \le M \le \min(N, 1000)$) — the number of features of objects excluding the target variable.

The following N lines contain a description of the objects. The i-th of these lines contains a description of the i-th object, $M+1$ integers. The first M of these integers: $X_{i,j}$ ($|X_{i,j}| \le 10^9$) are features of the i-th object, and the last Y_i ($|Y_i| \le 10^9$) is the target value.

Output

Print $M+1$ floating-point real numbers A_j — line coefficients from the linear equation $Y = A_0 \cdot X_0 + A_1 \cdot X_1 + \dots + A_{M-1} \cdot X_{M-1} + A_M$

Scoring

Let $Score = 100 \cdot \frac{B-S}{B-J}$ where S is SMAPE of your solution, J is SMAPE $\approx 1\%$, B is SMAPE $\approx 2\%$.

Then $\text{Verdict} = \begin{cases} \text{Ok} & \text{Score} \ge 100 \\ \text{PartiallyCorrect} & 0 \le \text{Score} < 100 \\ \text{WrongAnswer} & \text{Score} < 0 \end{cases}$.

input
```
2 1
2015 2045
2016 2076
```
output
```
31.0
-60420.0
```

input
```
4 1
1 0
1 2
2 2
2 4
```
output
```
2.0
-1.0
```

E. Support-vector machine

1 second, 256 megabytes

Find the coefficients λ_i of support vectors and the bias b, for classification according to the formula $class(x) = sign(\sum_i y_i \cdot \lambda_i \cdot k(x, x_i) + b)$, where x — is the vector description of the requested object, and k is the kernel function.

Input

The first line contains an integer N ($1 \le N \le 100$) — the number of objects in the training set.

The following N lines contain a description of the objects, one object per line. The i-th object is described by $N+1$ as an integer: the first N of them $K_{i,j}$ ($|K_{i,j}| \le 10^9$) is the value of the kernel function between the i-th and j-th objects, the last Y_i ($Y_i = \pm 1$) is the class of the i-th object.

Next is the line containing the integer C ($1 \le C \le 10^5$) — a restriction on the coefficients λ_i.

Output

Print $N+1$ floating-point numbers: first N numbers — coefficients λ_i ($0 \le \lambda_i \le C$, $\sum \lambda_i \cdot Y_i = 0$) corresponding to objects from the training set, the last number b ($|b| \le 10^{12}$) is a bias coefficient.

Scoring

Let $Score = 100 \cdot \frac{F-B}{J-B}$, where F is the F_1 measure of your solution, J is the F_1 measure of a standard solution with a margin of $\approx 1\%$, B is the F_1-measure of a naive solution with a margin of $\approx 2\%$.

Then $Verdict = \begin{cases} Ok & Score \ge 100 \\ PartiallyCorrect & 0 \le Score \le 100 \\ WrongAnswer & Score < 0 \end{cases}$

input
```
6
5 4 6 9 11 10 -1
4 5 6 9 10 11 -1
6 6 8 12 14 14 -1
9 9 12 18 21 21 1
11 10 14 21 25 24 1
10 11 14 21 24 25 1
1
```
output
```
0.0
0.0
1.0
1.0
0.0
0.0
-5.0
```

Statement is not available on English language

F. Наивный байесовский классификатор

1 секунда, 256 мегабайт

Реализуйте наивный байесовский классификатор.

Априорные вероятности классов оцениваются обыкновенным частотным методом.

Для оценки вероятности встречи слов в каждом классе используется модель Бернулли с аддитивным сглаживанием (сглаживание Лапласа) $p(x) = \frac{count(x)+\alpha}{\sum_{y \in Q} count(y) + \alpha \cdot |Q|}$, где x — рассматриваемое событие, а Q — множество всех событий.

Каждое слово это отдельный признак с двумя возможными событиями встретилось / не встретилось.

Входные данные

В первой строке содержится целое положительное число K ($1 \le K \le 10$) — число классов.

Во второй строке содержится K целых положительных чисел λ_C ($1 \le \lambda_C \le 10$) — штрафы за ошибки классификации сообщений соответствующих классов.

В третьей строке содержится целое положительное число α ($1 \le \alpha \le 10$) — интенсивность аддитивного сглаживания.

Следующая строка содержит целое положительное число N ($1 \le N \le 200$) — число сообщений в обучающей выборке.

Следующие N строк содержат описания соответствующих сообщений из обучающей выборки. Каждое сообщение в ней начинается с целого положительного числа C_i ($1 \le C_i \le K$) — класса к которому относится i-е сообщение. Далее следует целое положительное число L_i ($1 \le L_i \le 10^4$) — число слов в i-м сообщении. Затем следует содержание сообщения — L_i слов состоящих из маленьких латинских букв.

Далее в отдельной строке содержится целое положительное число M ($1 \le M \le 200$) — число сообщений в проверочной выборке.

Следующие M строк содержат описания соответствующих сообщений из проверочной выборки. Каждое сообщение в ней начинается с целого положительного числа L_j $(1 \le L_j \le 10^4)$ — число слов в j-м сообщении. Затем следует содержание сообщения — L_j слов состоящих из маленьких латинских букв.

Гарантируется, что сумма длин всех сообщений в обучающей и проверочной выборках меньше чем $2 \cdot 10^6$.

Выходные данные

Выведите M строк — результаты мягкой классификации оптимального наивного байесовского классификатора соответствующих сообщений из проверочной выборки. Допустимая абсолютная и относительная погрешность 10^{-4}.

Каждый j-й результат мягкой классификации должен содержать K чисел p_C — вероятности того, что j-е сообщение относится к классу C.

входные данные

```
3
1 1 1
1
4
1 2 ant emu
2 3 dog fish dog
3 3 bird emu ant
1 3 ant dog bird
5
2 emu emu
5 emu dog fish dog fish
5 fish emu ant cat cat
2 emu cat
1 cat
```

выходные данные

```
0.4869739479 0.1710086840 0.3420173681
0.1741935484 0.7340501792 0.0917562724
0.4869739479 0.1710086840 0.3420173681
0.4869739479 0.1710086840 0.3420173681
0.4869739479 0.3420173681 0.1710086840
```

В примере условные вероятности выглядят следующим образом:

$p(w_x \mid c_y)$	ant	bird	dog	emu	fish
c_1	$3/4$	$1/2$	$1/2$	$1/2$	$1/4$
c_2	$1/3$	$1/3$	$2/3$	$1/3$	$2/3$
c_3	$2/3$	$2/3$	$1/3$	$2/3$	$1/3$

Слово cat не рассматривается, так как оно ни разу не встретилось в обучающей выборке.

Для первого запроса X:

$p(c_1) \cdot p(X \mid c_1) = \frac{2}{4} \cdot \left(1 - \frac{3}{4}\right) \cdot \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{2}\right) \cdot \left(\frac{1}{2}\right) \cdot \left(1 - \frac{1}{4}\right)$ и

$p(c_1 \mid X) = \frac{3/256}{3/256 + 1/243 + 2/243}$

Statement is not available on English language

G. Дерево принятия решений

1.5 секунд, 256 мегабайт

Постройте дерево принятия решений.

Входные данные

Первая строка содержит три целых положительных числа M ($1 \le M \le 100$) — число признаков у объектов (исключая класс), K ($1 \le K \le 20$) — число классов и H $(1 \le H \le 10)$ — максимальная глубина (в рёбрах) дерева принятия решений.

Вторая строка содержит целое положительное число N $(1 \le N \le 4000)$ — число объектов в обучающей выборке.

Следующие N строк содержат описания объектов в обучающей выборке. В i-й из этих N строк перечислено $M + 1$ целое число: первые M чисел $A_{i,j}$ $(\left|A_{i,j}\right| \le 10^9)$ — признаки i-го объекта, последнее число C_i $(1 \le Ci \le K)$ — его класс.

Выходные данные

Выведите построенное дерево принятия решений.

В первой строке выведите целое положительное число S $(1 \le S \le 2^{11})$ — число вершин в дереве.

В следующих S строках выведите описание вершин дерева. В v-й из этих строк выведите описание v-й вершины:

- Если v-я вершина узел, выведите через пробел: заглавную латинскую букву 'Q', целое положительное число f_v $(1 \le f_v \le M)$ — индекс признака по которому происходит проверка в данном узле, вещественное число с плавающей точкой b_v — константа с которой происходит сравнения для проверки, два целых положительных числа l_v и r_v $(v < l_v, r_v \le S)$ — индекс вершины дерева в которую следует перейти, если выполняется условие $A[f_v] < b_v$, и индекс вершины дерева в которую следует перейти, если условие не выполняется.
- Если v-я вершина лист, выведите через пробел: заглавную латинскую букву 'C' и целое положительное число D_v $(1 \le D_v \le K)$ — класс объекта попавшего в данный лист.

Вершины нумеруются с единицы. Корнем дерева считается первая вершина.

Система оценки

Решение будет проверено на секретном наборе данных. На основании предсказанных и реальных классов вычисляется усреднённая по классам микро F_1-мера.

Пусть $Score = 100 \cdot \frac{F-B}{J-B}$, где F — F_1-мера вашего решения, J — F_1-мера решения эталона с запасом $\approx 1\%$, B — F_1-мера наивного решения с запасом $\approx 2\%$.

Тогда $Verdict = \begin{cases} Ok & Score \ge 100 \\ PartiallyCorrect & 0 \le Score \le 100 \\ WrongAnswer & Score < 0 \end{cases}$

входные данные

```
2 4 2
8
1 2 1
2 1 1
3 1 2
4 2 2
3 4 3
4 3 3
1 3 4
2 4 4
```

выходные данные

```
7
Q 1 2.5 2 5
Q 2 2.5 3 4
C 1
C 4
Q 2 2.5 6 7
C 2
C 3
```

Statement is not available on

English language

H. Логическое выражение

1 секунда❷, 256 мегабайт

Постройте искусственную нейронную сеть, вычисляющую логическую функцию f, заданную таблицей истинности.

Входные данные

Первая строка содержит целое число M $(1 \le M \le 10)$ — число аргументов f. Следующие 2^M строк содержат значения f в таблице истинности (0 — ложь, 1 — истина). Строки в таблице истинности последовательно отсортированы по аргументам функции от первого к последнему. Например:

$M=1$	$M=2$	$M=3$
$f(0)$	$f(0,0)$	$f(0,0,0)$
$f(1)$	$f(1,0)$	$f(1,0,0)$
	$f(0,1)$	$f(0,1,0)$
	$f(1,1)$	$f(1,1,0)$
		$f(0,0,1)$
		$f(1,0,1)$
		$f(0,1,1)$
		$f(1,1,1)$

Выходные данные

В первой строке выведите целое положительное число D $(1 \le D \le 2)$ — число слоёв (преобразований) в вашей сети.

На следующей строке выведите D целых положительных чисел n_i ($1 \le n_i \le 512$ и $n_D = 1$) — число искусственных нейронов на i-м слое. Предполагается, что $n_0 = M$.

Далее выведите описание D слоёв. i-й слой описывается n_i строками, описанием соответствующих искусственных нейронов на i-м слое. Каждый искусственный нейрон описывается строкой состоящей из n_{i-1} вещественных чисел с плавающей точкой w_j и одного вещественного числа b — описание линейной зависимости текущего нейрона от выходов предыдущего i-го слоя. Линейная зависимость задается по формуле: $Y = \sum w_j \cdot x_j + b$. Предполагается, что после каждого вычисления линейной зависимости к её результату применяется функция ступенчатой активации $a(Y) = \begin{cases} 1 & Y > 0 \\ 0 & Y < 0 \end{cases}$.

Обратите внимание, что в нуле данная функция не определена, и если в ходе вычисления вашей сети будет вызвана активация от нуля, вы получите ошибку.

входные данные

```
2
0
1
0
1
```

выходные данные

```
2
2 1
1.0 -1.0 -0.5
1.0 1.0 -1.5
1 1 -0.5
```

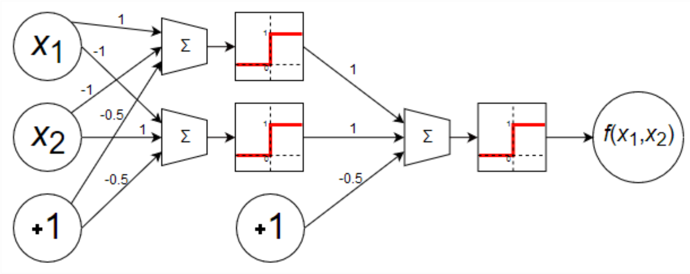
входные данные

```
2
0
1
1
0
```

выходные данные

```
2
2 1
1.0 -1.0 -0.5
-1.0 1.0 -0.5
1 1 -0.5
```

Во втором примере в результате получается следующая сеть:



Statement is not available on English language

I. Матричная функция

1 секунда❷, 256 мегабайт

Вычислите матричную функцию и её производную по заданному графу вычислений.

Входные данные

В первой строке содержится три целых положительных числа N, M, K $(1 \le M, K \le N \le 50)$ — число вершин в графе вычислений, число входных параметров (вершин) и число выходных параметров (вершин). Далее следует N строк — описание вершин графа вычислений. i-я из этих строк содержит описание i-й вершины:

- **var** r c $(1 \le r, c \le 25)$ — входной параметр функции, матрица состоящая из r строк и c столбцов.
- **tnh** x $(1 \le x < i)$ — матрица из значений гиперболического тангенса вычисленного от соответствующих компонент матрицы полученной из x-й вершины графа вычислений.
- **rlu** α^{-1} x $(1 \le \alpha^{-1} \le 100,\ 1 \le x < i)$ — матрица из значений функции параметрического линейного выпрямителя с параметром α вычисленной от соответствующих компонент матрицы полученной из x-й вершины графа вычислений. α^{-1} — целое число. Производная в нуле равна единице.
- **mul** a b $(1 \le a, b < i)$ — произведение матриц полученных из a-й b-й вершины графа вычислений соответственно.
- **sum** len u_1 u_2 ... u_{len} $(1 \le len \le 10,\ \forall_{1 \le j \le len}: 1 \le u_j < i)$ — сумма матриц полученных из вершин $u_1, u_2, \ldots, u_{len}$ графа вычислений.
- **had** len u_1 u_2 ... u_{len} $(1 \le len \le 10,\ \forall_{1 \le j \le len}: 1 \le u_j < i)$ — произведение Адамара (покомпонентное) матриц полученных из вершин $u_1, u_2, \ldots, u_{len}$ графа вычислений.

Гарантируется, что первые M вершин и только они имеют тип **var**. Последние K вершин считаются выходными. Гарантируется, что размеры матриц аргументов для каждой вершины согласованны.

Далее следует описание M матриц — входных параметров соответствующих вершин графа вычислений в порядке возрастания их индексов.

Затем следует описание K матриц — производных функции по соответствующим выходным вершинам в порядке возрастания их индексов. Обратите внимание, что производные вычислены только из некоторых скрытых вершин. Если какая-та выходная вершина зависит от другой выходной вершины, то соответствующую производную нужно досчитать.

Каждая строка, каждой матрицы расположена на отдельной строке. Матрицы состоят из целых чисел по модулю не превышающих 10.

Выходные данные

Выведите K матриц — значение параметров соответствующих выходных вершин графа вычисления в порядке возрастания их индексов. Затем выведите M матриц производных функции по соответствующим входным вершинам в порядке возрастания их индексов. Допустимая абсолютная и относительная погрешность 10^{-4}.

входные данные

```
6 3 1
var 1 3
var 3 2
var 1 2
mul 1 2
sum 2 4 3
rlu 10 5
-2 3 5
4 2
-2 0
2 1
4 -2
-1 1
```

выходные данные

```
0.0 -0.1
-3.8 2.0 -1.9
2.0 -0.2
-3.0 0.3
-5.0 0.5
-1.0 0.1
```

В примере вычисляется функция

$$ReLU_{\alpha=0.1}\left(\begin{pmatrix}-2 & 3 & 5\end{pmatrix} \times \begin{pmatrix}4 & 2\\ -2 & 0\\ 2 & 1\end{pmatrix} + \begin{pmatrix}4 & -2\end{pmatrix}\right), \text{ а } \begin{pmatrix}-1 & 1\end{pmatrix}$$

производная по её выходу.

Statement is not available on English language

J. Свёрточная сеть

1 секунда❓, 256 мегабайт

Посчитайте значение выхода свёрточной сети и пересчитайте её производную.

Входные данные

В первой строке содержится описание входа свёрточной сети, трёхмерной матрицы. Высота этой матрицы совпадает с её шириной. Первое число N_0 $(1 \le N_0 \le 40)$ — высота и ширина входной трёхмерной матрицы, второе число D_0 $(1 \le D_0 \le 10)$ — её глубина. Следующие $D_0 \times N_0 \times N_0$ чисел — описание трёхмерной матрицы, значения её ячеек выписанных в порядке: глубина, высота, ширина.

Следующая строка содержит одно число L $(1 \le L \le 10)$ — число слоёв (преобразований) в сети.

Следующие L строк содержат описания соответствующих преобразований:

- **relu** α^{-1} $(1 \le \alpha^{-1} \le 100)$ — функции параметрического линейного выпрямителя с параметром α.
- **pool** S $(1 \le S \le 5)$ — операция субдискретизации (подвыборки) по высоте и ширине размера $S \times S$ с шагом S. В качестве свёртки используется операция максимума. Производная для максимума вычисляется как: $\frac{\partial \max}{\partial x_i}(x) = 1$ если $x_i = \max(x)$, иначе 0.
- **bias** $B_1, B_2, \ldots, B_D$ $(|B_i| \le 10)$ — операция сдвига, прибавляющая к каждой ячейке матрицы на глубине i значение B_i, D — глубина матрицы до и после преобразования.
- **cnvm** $H\ K\ S\ P\ A_{1,1,1,1}, A_{1,1,1,2}, \ldots, A_{H,D,K,K}$ $(1 \le H \le 10,\ 1 \le K \le 5,\ 1 \le S \le K,\ 0 \le P < K,\ |A_i| \le 10)$ — свёртка с ядром A размера $H \times D \times K \times K$ с шагом S с зеркальным заполнением рамки размера P, где D — глубина матрицы до преобразования. H — глубина матрицы после преобразования. Значения ячеек A выписаны в порядке: глубина полученной матрицы, глубина исходной матрицы, высота ядра, ширина ядра.
- **cnve** $H\ K\ S\ P\ A_{1,1,1,1}, A_{1,1,1,2}, \ldots, A_{H,D,K,K}$ — свёртка с расширением границы. Аналогична предыдущей.
- **cnvc** $H\ K\ S\ P\ A_{1,1,1,1}, A_{1,1,1,2}, \ldots, A_{H,D,K,K}$ — свёртка с заполнением с циклическим сдвигом. Аналогична предыдущей.

Гарантируется, что размеры всех многомерных матриц согласованы с соответствующими гипер-параметрами преобразований.

В последней строке записана производная по выходу сети.

Все числа во входных данных целые.

Выходные данные

Выведите значение выходной трёхмерной матрицы.

Далее выведите производную по входу сети.

Затем для каждого слоя сдвига и свёртки в возрастающем порядке номера слоя выведите производную по его параметрам.

Выходные матрицы могут содержать числа с плавающей точкой. Допустимая абсолютная и относительная погрешность 10^{-4}.

входные данные

```
4 1 4 3 2 1 3 2 1 0 2 1 0 1 1 0 1 2
4
cnvm 1 3 3 1 0 -1 0 -1 0 -1 0 -1 0
bias 4
relu 8
pool 2
1
```

выходные данные

```
0.0
0.0 0.0 -2.0 0.0 0.0 0.0 0.0 -2.0 -2.0 0.0 0.0 -2.0 0.0 -2.0 -2.0
0.0
2.0 3.0 2.0 3.0 4.0 3.0 2.0 3.0 2.0
3.0
```

Пример заполнения угла рамки для свёрточного слоя:

cnvm							
18	17	16	15	16	17	18	19
13	12	11	10	11	12	13	14
8	7	6	5	6	7	8	9
3	2	1	0	1	2	3	4
8	7	6	5	6	7	8	9
13	12	11	10	11	12	13	14
18	17	16	15	16	17	18	19
23	22	21	20	21	22	23	24

cnve							
0	0	0	0	1	2	3	4
0	0	0	0	1	2	3	4
0	0	0	0	1	2	3	4
0	0	0	0	1	2	3	4
5	5	5	5	6	7	8	9
10	10	10	10	11	12	13	14
15	15	15	15	16	17	18	19
20	20	20	20	21	22	23	24

cnvc							
12	13	14	10	11	12	13	14
17	18	19	15	16	17	18	19
22	23	24	20	21	22	23	24
2	3	4	0	1	2	3	4
7	8	9	5	6	7	8	9
12	13	14	10	11	12	13	14
17	18	19	15	16	17	18	19
22	23	24	20	21	22	23	24

Statement is not available on English language

K. LSTM сеть

1 секунда❓, 256 мегабайт

Дана сеть LSTM для обработки последовательностей.

Каждый блок этой сети вычисляет результат по формулам: $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$, $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$, $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$, $c_t = f_t \circ c_{t-1} + i_t \circ tanh(W_c x_t + U_c h_{t-1} + b_c)$ и $h_t = o_t \circ c_t$, где x_t — вход t-го блока, h_t и c_t — векторы краткосрочной и долгосрочной памяти, o_t — выход t-го блока, а $\circ$ – произведение Адамара.

Входные данные

В первой строке находится число N ($1 \le N \le 20$) — размер векторов LSTM.

Далее перечислены соответствующие матрицы и вектора W_f, U_f, B_f, W_i, U_i, B_i, W_o, U_o, B_o, W_c, U_c, B_c.

Затем следует число M ($1 \le M \le 20$) — число элементов последовательности обрабатываемой LSTM сетью.

Далее следуют два вектора h_0 и c_0, а также M векторов x_t.

Затем следует вектора производных сети по выходным векторам h_M и c_M, а также M векторов производных по выходам o_t **в обратном порядке** $o_M, o_{M-1}, \ldots, o_1$.

Все вектора записаны N числами разделёнными пробелами на отдельной строке, а матрицы N векторами размера N. Все элементы векторов и матриц целые числа по модулю не превосходящие 10.

Выходные данные

Сперва выведите M векторов выходов сети o_t.

Далее выведите два последних вектора памяти h_M и c_M.

Затем выведите M векторов производных сети по входам x_t **в обратном порядке**.

Далее выведите два вектора производных сети по h_0 и c_0.

После выведите производные по соответствующим матрицам и векторам параметров LSTM: W_f, U_f, B_f, W_i, U_i, B_i, W_o, U_o, B_o, W_c, U_c, B_c.

Выходные вектора и матрицы могут содержать числа с плавающей точкой. Допустимая абсолютная и относительная погрешность 10^{-6}.

входные данные

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

выходные данные

```
1.233945759863131E-4
-2.875857041962763E-5
-0.23306186831759548
-0.37692699674663843
0.21113860108361812
-0.047420021082055105
0.27102651105684017
0.13551325552842008
0.13551325552842008
0.159905268234481
0.0799526341172405
0.0799526341172405
1.8924865599381104E-4
9.462432799690552E-5
9.462432799690552E-5
-0.10011198258925587
-0.050055991294627934
-0.050055991294627934
```

L. Pearson correlation coefficient

1 second, 256 megabytes

Count the Pearson correlation coefficient of two numerical features.

Input

The first line contains a positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of of these N lines contains a description of one object: two integers x_1 and x_2 ($-10^9 \le x_1, x_2 \le 10^9$) are the values of the first and second features of the described object.

Output

Print one real floating-point number — the value of Pearson correlation coefficient of two features for given objects.

input

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

output

```
-0.500000000
```

M. Spearman's rank correlation coefficient

1 second, 256 megabytes

Count the Spearman's rank correlation coefficient of two numerical features.

Input

The first line contains a positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of of these N lines contains a description of one object: two integers x_1 and x_2 ($-10^9 \le x_1, x_2 \le 10^9$) are the values of the first and second features of the described object. It is guaranteed that all values of each feature are different.

Output

Print one real floating-point number — the value of Spearman's rank correlation coefficient of two features for given objects.

input

```
5
1 16
2 25
3 1
4 4
5 9
```

output

```
-0.500000000
```

N. Distances

1 second, 256 megabytes

Calculate the dependence of the categorical feature Y on the numerical X by intraclass and interclass distance:

- Intraclass distance $= \sum_{i,j:y_i=y_j} \left|x_i - x_j\right|$
- Interclass distance $= \sum_{i,j:y_i \neq y_j} \left|x_i - x_j\right|$

Input

The first line contains a single positive integer K ($1 \le K \le 10^5$) — the maximum number of different values Y of the second feature.

The next line contains a single positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of these N lines contains a description of one object: two integers x and y ($|x| \le 10^7, 1 \le y \le K$) — the values of the first and second feature described object.

Output

On the first line, print a single integer — the intraclass distance.

In the second line print a single integer — interclass distance.

input

```
2
4
1 1
2 2
3 2
4 1
```

output

```
8
12
```

O. Conditional variance

1 second, 256 megabytes

Calculate the conditional variance of $D(Y|X)$.

Input

The first line contains a single positive integer K ($1 \le K \le 10^5$) — the maximum number of different values of the feature X.

The next line contains a positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of these N lines contains a description of one object: two positive integers x and y ($1 \le x \le K$, $|y| \le 10^9$) — values of features X and Y.

Output

Print a single floating-point number — conditional variance.

input

```
2
4
1 1
2 2
2 3
1 4
```

output

```
1.25
```

P. Chi-square

1 second, 256 megabytes

Calculate the dependence of two categorical signs according to the Pearson's chi-squared test.

Input

The first line contains two positive integers K_1 and K_2 ($1 \le K_1, K_2 \le 10^5$) — the maximum number of different values of the first and second feature.

The next line contains a positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of these N lines contains a description of one object: two positive integers x_1 and x_2 ($1 \le x_1 \le K_1$, $1 \le x_2 \le K_2$) are the values of the first and second feature of the described object.

Output

Print a single floating-point number — the chi-square test of the dependence of two features on given objects.

input

```
2 3
5
1 2
2 1
1 1
2 2
1 3
```

output

```
0.833333333
```

In the example, the actual number of observations of observations looks like $\begin{array}{cccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{1} & 1 & 1 & 1 \\ \mathbf{2} & 1 & 1 & 0 \end{array}$, while the expected number of observations $\begin{array}{cccc} & \mathbf{1} & \mathbf{2} & \mathbf{3} \\ \mathbf{1} & 1.2 & 1.2 & 0.6 \\ \mathbf{2} & 0.8 & 0.8 & 0.4 \end{array}$.

Q. Conditional entropy

1 second, 256 megabytes

Count the conditional entropy of $H(Y|X)$. For calculations, use the natural logarithms $ln(x) = \log_e(x)$.

Input

The first line contains two positive integers K_x and K_y ($1 \le K_x, K_y \le 10^5$) — the maximum number of different values of the attributes X and Y.

The next line contains a positive integer N ($1 \le N \le 10^5$) — the number of objects.

The following N lines contain descriptions of the corresponding objects. Each of these N lines contains a description of one object: two positive integers x and y ($1 \le x \le K_x$, $1 \le y \le K_y$) — values of the X and Y.

Output

Print a single floating-point number conditional entropy.

input

```
2 3
5
1 2
2 1
1 1
2 2
1 3
```

output

```
0.9364262454248438
```