

Задача 1. Матрица ошибок

При оценке качества бинарной классификации часто используют матрицу ошибок (Confusion matrix). Поясним, что это такое. Пусть у нас есть N каких-то объектов, относящихся к одному из двух классов (0 или 1). Например, объекты — это рентгеновские снимки, класс 1 означает, что пациент болен, класс 0 — что здоров. Пусть имеется некоторый алгоритм, который по объекту предсказывает его класс. Зная фактические и предсказанные классы, для оценки качества предсказания можно построить следующую таблицу:

	Предсказанный класс = 1	Предсказанный класс = 0
Фактический класс = 1	TP	FN
Фактический класс = 0	FP	TN

Здесь значение TP (True Positive) — это количество объектов, для которых модель предсказала класс 1, и у них на самом деле класс 1. Значение TN (True Negative) — количество объектов, для которых модель предсказала класс 0, и у них на самом деле класс 0. Значение FP (False Positive) — количество объектов, для которых модель предсказала класс 1, а на самом деле у них класс 0. Значение FN (False Negative) — количество объектов, для которых модель предсказала класс 0, а на самом деле у них класс 1.

По значениям из данной таблицы можно вычислить ещё две характеристики — точность и полноту.

Точность (precision) вычисляется по формуле $precision = TP / (TP + FP)$, то есть какая доля объектов, которые наш алгоритм отнёс к классу 1, действительно относится к классу 1.

Полнота (recall) вычисляется по формуле $recall = TP / (TP + FN)$, то есть какую долю объектов класса 1 среди всех объектов класса 1 нашёл наш алгоритм.

Пусть всего было 99 объектов, из них 35 — класса 0 и 64 — класса 1. Известно, что модель правильно определила класс для 51 объекта. Найдите ответы на следующие четыре вопроса.

1. Какая максимальная точность могла получиться?
2. А какая минимальная?
3. Пусть дополнительно известно, что точность оказалась равна полноте. Какая максимальная полнота при этом могла получиться?
4. А какая минимальная?

В ответ запишите четыре вещественных числа, каждое в отдельной строке. В качестве разделителя дробной части используйте точку. Если вы не знаете каких-то ответов, напишите вместо них нули. Гарантируется, что все числа в ответе имеют конечную дробную часть.

Замечание

Система оценивания: каждый верный ответ оценивается в 25 баллов.

Задача 2. Результаты олимпиады

Вам дана таблица с данными о проверке решений участников некоторой олимпиады по информатике: <https://olympiads.vogu35.ru/school/informatics/2024-2025/log.csv>. Копия этой таблицы также находится здесь: <https://avt.vogu35.ru/files/olymp/2024-2025/log.csv>.

Таблица имеет 4 столбца: problem — номер задачи (от 1 до 5); userId — идентификатор участника; language — язык программирования; score — баллы, которое набрало решение (от 0 до 100). Значения score также могут быть пустыми (если, например, какие-то решения даже не скомпилировалось на сервере).

Олимпиада проводилась по традиционным школьным правилам: участник может послать на проверку несколько решений по каждой задаче, при подведении итогов выбирается лучшее решение участника по каждой задаче, баллы за них складываются.

Проанализируйте данные в таблице и ответьте на следующие 4 вопроса.

1. Сколько участников набрали больше нуля баллов?
2. Сколько участников все свои решения отправили на языке Python?
3. Сколько участников набрали не менее 200 баллов?
4. Сколько участников, набравших не менее 200 баллов, хотя бы одно решение отправили на языке C++?

В ответе напишите 4 целых числа, каждое в отдельной строке. Не пишите ничего лишнего. Если вы не знаете каких-то ответов, напишите вместо них нули.

Замечание

Каждый верный ответ оценивается в 25 баллов.

Задача 3. Верный диагноз

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Известная медицинская клиника решила добавить в свой список услуг лечение опасной болезни под названием «Воспаление хитрости». Но, прежде чем лечить болезнь, её необходимо диагностировать. Для этого клиника заказала у трёх медицинских лабораторий разработку тест-систем для выполнения диагностики. Через некоторое время лаборатории предоставили свои результаты.

Известно, что при своей работе все тест-системы используют разные технологии и разные алгоритмы для выявления болезни. Поэтому в этой задаче будем считать их диагнозы независимыми друг от друга (*примечание: на практике некоторая зависимость всё равно бы имела, но здесь её будем игнорировать*).

Первая тест-система ставит верный диагноз с вероятностью p_1 , вторая — с вероятностью p_2 , третья — с вероятностью p_3 . Окончательное решение принимается по большинству результатов. То есть, если любые две или все три тест-системы решили, что пациент болен, то он считается больным, в противном случае — нет. Определите, с какой вероятностью клиника будет ставить верные диагнозы.

Формат входных данных

Вводятся три вещественных числа p_1 , p_2 , p_3 , каждое в отдельной строке ($0.5 \leq p_1, p_2, p_3 \leq 1$).

Формат выходных данных

Выведите одно вещественное число — итоговую вероятность верного диагноза. Абсолютная или относительная погрешность ответа не должна превышать 10^{-4} .

Пример

стандартный ввод	стандартный вывод
0.65 0.7 0.9	0.851

Задача 4. Меры центральной тенденции

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Меры центральной тенденции — это статистические показатели, которые описывают «типичное» или центральное значение в наборе данных. Существует несколько таких мер, рассмотрим три из них.

Медианой набора чисел называется элемент, который окажется в середине, если числа отсортировать. **Мода** — это самый часто встречающийся элемент. Мод может быть несколько. **Среднее арифметическое** — это сумма всех элементов, делённая на их количество.

Напишите программу, находящую набор из пяти целых чисел, для которого выполняются одновременно следующие три условия:

- мода равна x , и она единственная;
- медиана равна y ;
- среднее арифметическое равно z .

Формат входных данных

Вводятся три целых числа x , y и z , каждое в отдельной строке ($0 \leq x, y, z \leq 10^8$).

Формат выходных данных

Выведите пять целых чисел, имеющих указанные моду, медиану и среднее. Числа должны быть в диапазоне от -10^9 до 10^9 . Если есть несколько верных ответов, выведите любой. Если решений нет, выведите одно число -1.

Примеры

стандартный ввод	стандартный вывод
1 2 5	1 9 1 2 12
1 3 2	-1

Замечание

Решения, верно работающие при $x, y, z \leq 10$, будут оценены в 50 баллов.

Задача 5. Ближайшие соседи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Даны n точек на плоскости. Каждая точка имеет координаты x_i, y_i и цвет c_i , который может быть белый ($c_i = 0$) или чёрный ($c_i = 1$).

Чтобы определить цвет новой точки P , если он неизвестен, может использоваться следующий подход. Берутся k наиболее близких к ней наших точек (где k — некоторое нечётное число), и выбирается тот цвет, который встречается среди них чаще. Примечание: меру близости двух точек можно вычислять по-разному, в данной задаче используется обычное евклидово расстояние (то есть длина отрезка между точками).

Качество работы такого классификатора зависит от выбора параметра k . Чтобы найти хорошее значение k , можно использовать следующий простой способ. Посмотрим, какие ответы при разных k будут получаться для наших входных точек (для которых заранее известны правильные ответы), и выберем то значение k , при котором правильных ответов будет наибольшее количество. Такой метод называется «LOO-кросс-валидация».

Поясним работу этого метода подробнее. Переберём все нечётные значения k в диапазоне от 1 до $n - 1$. Для каждого такого k переберём все точки. Для каждой из них найдём k ближайших к ней других точек (её саму не учитываем), возьмём цвет, который встречается среди них чаще, и сравним его с настоящим цветом этой точки. Чем больше верных ответов получилось для текущего значения k , тем это значение лучше. Разумеется, для повышения эффективности можно изменить описанную реализацию алгоритма — главное, чтобы при этом результаты остались правильными.

Напишите программу, которая для каждого нечётного k в диапазоне от 1 до n определит, для скольких точек будет верно определён цвет при данном k в ходе LOO-кросс-валидации.

Формат входных данных

В первой строке входных данных записано целое число n ($2 \leq n \leq 1000$).

В следующих n строках записаны тройки чисел x_i, y_i и c_i — координаты очередной точки ($-20000 \leq x_i, y_i \leq 20000$) и её цвет c_i (равный либо 0, либо 1).

Гарантируется, что никакие две точки не совпадают и расстояния между всеми парами точек различны.

Формат выходных данных

Выведите количество верно классифицированных точек для $k = 1, k = 3, k = 5$ и так далее до $n - 1$ (а если $n - 1$ чётно, то до $n - 2$). Сами значения k выводить не нужно.

Пример

стандартный ввод	стандартный вывод
4	2
1 2 0	0
-3 1 1	
0 0 1	
4 5 0	

Замечание

Подзадача 1 (до 30 баллов): $n \leq 50$.

Подзадача 2 (до 30 баллов): $n \leq 200$.

Подзадача 3 (до 40 баллов): $n \leq 1000$.

Примечание для пишущих на языке Python. Ввести три числа через пробел можно так:

```
x, y, c = [int(x) for x in input().split()]
```