

## Максимальное количество баллов за олимпиаду — 600

## Задание 1. Задача по геометрии

Саша нарисовал выпуклый многоугольник с помощью компьютерной программы и отправил модели ИИ запрос вычислить сумму его углов. Он получил неверный ответ  $500^\circ$ . Оказалось, что один из углов многоугольника модель учла дважды. Чему равен этот угол? Ответ выразите в градусах. Напомним, что в выпуклом многоугольнике все углы меньше  $180^\circ$ .

**Ответ:** 140

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

**Решение.** Если в Сашином многоугольнике хотя бы пять вершин, то сумма его углов хотя бы  $180^\circ \cdot 3 = 540^\circ$ , этот случай невозможен. Если это треугольник, сумма углов равна  $180^\circ$ , и учтенный дважды угол должен быть больше  $180^\circ$ , что невозможно. Значит, у Саши четырехугольник, его сумма углов равна  $360^\circ$ , и дважды был посчитан угол величиной  $500^\circ - 360^\circ = 140^\circ$ .

## Задание 2. Математика в чат-боте

Дима выбрал два натуральных числа  $a$  и  $b$  и затем отправил запрос модели ИИ найти значение выражения  $a + b \cdot 2$  (он записал это выражение на бумажке, сфотографировал и загрузил полученную фотографию). Из-за неаккуратного почерка модель распознала записанное выражение как  $a + b^2$  и в результате ответ модели оказался на 80 больше правильного. Найдите последнюю цифру числа  $a \cdot b$ .

**Ответ:** 0

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

**Решение.** Исходя из условия, мы получаем, что  $b^2 - 2b = 80$ , то есть  $(b - 10)(b + 8) = 0$ . Следовательно,  $b = 10$ , поскольку число  $b$  — натуральное, а тогда  $ab$  оканчивается нулём.

## Задание 3. Градиентный спуск

На прямой изучают работу очень простого «искусственного интеллекта», который зависит всего от одного числа — параметра  $w$ . Для каждого целого  $w$  от 1 до 10 заранее посчитана ошибка  $E(w)$  этого ИИ на обучающих примерах:

$w$	1	2	3	4	5	6	7	8	9	10
$E(w)$	9	5	2	4	6	7	3	1	2	4

ИИ обучают с помощью следующего алгоритма изменения параметра  $w$  (аналог градиентного спуска).

1. Сначала выбирают начальное целое значение параметра  $w$  от 1 до 10.
2. Рассматривают «соседей» текущего значения  $w$ :
  - слева — число  $w - 1$  (если  $w > 1$ );
  - справа — число  $w + 1$  (если  $w < 10$ ).
3. Если среди существующих соседей есть такие, у которых ошибка строго меньше:  $E(w_{\text{сосед}}) < E(w)$ , то переходят к тому соседу, у которого ошибка наименьшая (среди соседей).
4. Затем снова выполняют шаг 2 и так далее, пока не окажется, что у всех существующих соседей ошибка не меньше текущей. В этот момент алгоритм останавливается; говорят, что он застрял в локальном минимуме.

Из таблицы видно, что наименьшее значение ошибки достигается при  $w = 8$ ; это глобальный минимум.

В реальных задачах часто запускают обучение много раз из разных начальных точек, чтобы увеличить шанс попасть в глобальный минимум. Будем считать, что:

- в каждом запуске начальное значение  $w$  выбирается случайно и равновероятно из чисел 1, 2, ..., 10;
- разные запуски независимы;
- запуск считается успешным, если алгоритм в итоге остановился в глобальном минимуме при  $w = 8$ .

При каком наименьшем количестве запусков вероятность того, что хотя бы один запуск окажется успешным, будет не меньше 95%?

**Ответ:** 5

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

**Решение.** По таблице видно, что слева все траектории спуска (из  $w = 1, 2, 3, 4, 5$ ) приходят в локальный минимум при  $w = 3$ , а справа все траектории (из  $w = 6, 7, 8, 9, 10$ ) приходят в глобальный минимум при  $w = 8$ .

Значит, при случайном равновероятном выборе начального  $w$  из  $\{1, \dots, 10\}$  вероятность успешного запуска (то есть попадания в  $w = 8$ ) равна

$$p = \frac{5}{10} = \frac{1}{2}.$$

При  $N$  независимых запусках вероятность того, что *хотя бы один* из них успешен, равна

$$1 - (1 - p)^N = 1 - \left(\frac{1}{2}\right)^N.$$

Нужно, чтобы

$$1 - \left(\frac{1}{2}\right)^N \geq 0,95 \iff \left(\frac{1}{2}\right)^N \leq 0,05.$$

Подбором:

$$\left(\frac{1}{2}\right)^4 = \frac{1}{16} \approx 0,0625 > 0,05, \quad \left(\frac{1}{2}\right)^5 = \frac{1}{32} \approx 0,03125 < 0,05.$$

Минимальное  $N$ , при котором условие выполняется, равно 5.

## Задание 4. Лидер продаж в категории

Интернет-магазин агрегирует товары по категориям. Данные находятся в файле, который вы можете скачать в форматах [XLSX](#), [ODS](#) или [CSV](#).

Для каждого товара известны три значения: `category`, `score`, `label`.

- `category` — категория товара;
- `score` — оценка интереса, число из диапазона  $[0, 1]$ ;
- `label` — факт покупки: 0 или 1.

В каждой категории на витрине показываются все товары, у которых значение `score` является максимальным среди всех товаров той же категории. Для каждой категории найдите её максимальную оценку:

`max_score(g) = max{score : category = g}`.

Выберите все строки, где `score = max_score(g)`. Если в категории несколько товаров делят максимум, выбираются все. Сколько выбранных строк имеют `label = 1`?

**Ответ:** 49

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

**Решение.** В каждой категории  $g$  находим максимальную оценку интереса

$$\text{max\_score}(g) = \max\{\text{score} : \text{category} = g\},$$

и выбираем все строки, где `score = max_score(g)`. Ответ — сумма значений `label` среди выбранных строк.

**Код (pandas):**

```
import pandas as pd

df = pd.read_csv("task9_group_top_select.csv")

max_score = df.groupby("category")["score"].transform("max")
sel = df["score"] == max_score

answer = df.loc[sel, "label"].sum()
print(answer)
```

## Задание 5. ICPC

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

В машинном обучении ансамбли работают лучше, когда в них есть разнообразие моделей: смешивают разные архитектуры и источники признаков, чтобы усилить общий результат. Однородные ансамбли часто переобучаются и хуже обобщают, а разнородные — устойчивее и сильнее.

Рассмотрим команды как ансамбль людей. Скоро начнётся новый сезон ICPC, а значит, известному тренеру Михаилу необходимо собрать команду, которая его выиграет! Ранее такое уже случалось, получится и в этом году. Секрет прост — нужно, чтобы в команде были и математики, и программисты. Если в команду войдут только программисты

или только математики, результат хорошим не будет. Прямо сейчас у Михаила для распределения есть  $n$  математиков и  $m$  программистов.

Сколькими способами можно собрать ровно одну команду из трёх человек на ICPC?

## Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^5$ ) — количества математиков и информатиков соответственно.

## Формат выходных данных

Выведите одно целое число — количество способов собрать одну успешную команду из трёх человек для участия в ICPC.

## Примеры

стандартный ввод	стандартный вывод
2	9
3	

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

## Решение

Есть два непересекающихся способа собрать команду: два математика + программист и два программиста + математик.

Посчитаем общую формулу: сколько есть способов собрать команду, если двух людей нужно выбрать из группы из  $a$  человек, а одного человека нужно выбрать из группы из  $b$  человек. Двух людей можно выбрать  $a(a-1)/2$  способами, одного человека —  $b$  способами. Для каждого выбора пары подходит любой выбор одиночного участника, значит всего способов  $a(a-1)/2 \cdot b$ .

Применяя это к нашей задаче, получаем итоговый ответ:  $n(n-1)/2 \cdot m + m(m-1)/2 \cdot n$ .

```
#include <iostream>
#include <stdint>
using namespace std;

int main() {
    int64_t n, m;
    cin >> n >> m;
    cout << m * n * (n - 1) / 2 + n * m * (m - 1) / 2;
    return 0;
}
```

## Задание 6. Коллектив

Ограничение по времени: 2 секунды

Ограничение по памяти: 256 мегабайт

В эпоху больших данных обучение становится распределённым, а ресурсы — на вес золота. В случае распределённого обучения на разных устройствах какие-то операции возможно производить эффективно, только если необходимые модули находятся в одном кластере. В таких случаях очень важно уметь распределять вычисления для достижения наилучшего результата с данными ресурсами.

Эту идею можно продемонстрировать на взаимодействии людей. В исследовательском центре одной небезызвестной компании работает  $n$  сотрудников. Про каждого сотрудника известно, что он является выпускником вуза с номером  $b_i$ , а его личный вклад в решение сложных задач равен  $a_i$ .

Сотрудникам необходимо решить очень сложную задачу. Для большей эффективности они будут работать над задачей парами. Рассмотрим все пары различных сотрудников  $(i, j)$ , где  $1 \leq i < j \leq n$ :

- пара сотрудников  $(i, j)$  считается допустимой, если они выпускники одного и того же вуза, то есть  $b_i = b_j$ ;
- вклад допустимой пары  $(i, j)$  в решение равен  $a_i + a_j$ ;
- если  $b_i \neq b_j$ , то такая пара не является допустимой и не даёт вклада в решение.

Каждый работник может входить в несколько допустимых пар с разными коллегами. При этом каждая конкретная пара сотрудников  $(i, j)$  учитывается не более одного раза. Требуется посчитать суммарный вклад всех допустимых пар сотрудников.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $n$  ( $2 \leq n \leq 10^5$ ) — количество сотрудников.

Вторая строка содержит  $n$  целых чисел  $a_1, a_2, \dots, a_n$ , где  $a_i$  ( $1 \leq a_i \leq 10^6$ ) — вклад  $i$ -го сотрудника.

Третья строка содержит  $n$  целых чисел  $b_1, b_2, \dots, b_n$ , где  $b_i$  ( $1 \leq b_i \leq 10^6$ ) — номер вуза, который окончил  $i$ -й сотрудник.

### Формат выходных данных

Выведите одно целое число — суммарный вклад всех допустимых пар сотрудников.

### Замечание

В первом примере есть только один выпускник вуза 1, поэтому он не сможет образовать ни одной пары. При этом есть 3 выпускника вуза 2 и они образуют пары, которые дадут вклады  $4 + 8$ ,  $4 + 8$ ,  $4 + 4$  в решение. Суммарный вклад будет  $12 + 12 + 8 = 32$ .

### Примеры

стандартный ввод	стандартный вывод
4	32
4 8 1 4	
2 2 1 2	

**Критерий оценивания:** точное совпадение ответа — 100 баллов

**Максимальный балл за задание — 100**

### Решение

В итоговый результат вклад дают только пары с одинаковым  $b$ . Давайте для каждого значения  $b$  посчитаем  $\text{cnt}[b]$  — количество людей с таким  $b$ .

Заметим, что каждый человек  $i$  образует допустимую пару ровно с  $\text{cnt}[b_i] - 1$  людьми. В каждую такую пару он даёт вклад  $a_i$  (вторую половину вклада даёт другой человек), поэтому суммарный вклад человека  $i$  в ответ равен  $(\text{cnt}[b_i] - 1) \cdot a_i$ . Ответ получается суммированием этой величины по всем людям.

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    int n;
    cin >> n;
    vector<long long> a(n), b(n);
    for (int i = 0; i < n; ++i) cin >> a[i];
    for (int i = 0; i < n; ++i) cin >> b[i];

    const int MAXB = 1000000;
    vector<long long> cnt(MAXB + 1, 0);
    for (int i = 0; i < n; ++i) ++cnt[b[i]];

    long long ans = 0;
    for (int i = 0; i < n; ++i) {
        ans += a[i] * (cnt[b[i]] - 1);
    }

    cout << ans << '\n';
    return 0;
}
```