

Большие вызовы 2023. Большие данные, искусственный интеллект, финансовые технологии и машинное обучение

8:00—12:45 15 апр 2023 г.

№ 1 — 2

0.25 баллов

Вычислите значение следующего выражения:

$$\sqrt[3]{20 + \sqrt{392}} + \sqrt[3]{20 - \sqrt{392}}.$$

Число или дробь

0.25 баллов

Вычислите значение следующего выражения:

$$\sqrt[3]{5\sqrt{2} + 7} - \sqrt[3]{5\sqrt{2} - 7}.$$

Число или дробь

№ 3 — 4

0.25 баллов

Найдите последнюю цифру числа $1 + 3 + 5 + 7 + 9 + \dots + 2011 + 2013$.

Число

0.25 баллов

Найдите последнюю цифру числа $1 \cdot 3 \cdot 5 \cdot 7 \cdot 9 \cdot \dots \cdot 2011 \cdot 2013$.

Число

№ 5

0.5 баллов

На плоскости нарисован правильный **30**-угольник. Сколько существует различных правильных **5**-угольников, вершины которых лежат в вершинах исходного **30**-угольника.

Число

№ 6

0.5 баллов

В классе учится **13** мальчиков и **11** девочек. Сколько существует способов сформировать команду из 4 учеников этого класса, чтобы в ней было 2 мальчика и 2 девочки?

Число

№ 7

0.5 баллов

Найдите сумму всех пятизначных натуральных чисел.

Число

№ 8

0.5 баллов

Петя выбрал случайное трёхзначное натуральное число. Какова вероятность, что сумма его цифр равна **9**?

Число или дробь

№ 9 — 10

0.25 баллов

Найдите наибольший общий делитель $10^{27} - 1$ и $10^{18} - 1$.

Ответ представьте в виде десятичной записи.

Число

0.25 баллов

Найдите наибольший общий делитель $10^{27} + 1$ и $10^{18} + 1$.

Ответ представьте в виде десятичной записи.

Число

№ 11

0.5 баллов

Одной кастрюли супа достаточно, чтобы накормить или **2** взрослых, или **5** детей. Есть **5** кастрюль супа и **15** голодных детей. Сколько взрослых можно накормить тем, что останется после того, как поедят дети?

Число

№ 12

0.5 баллов

Найдите все натуральные n , не превосходящие **300**, для которых значение выражения $n^2 - 16n + 63$ делится на **143**.

Число



№ 13

0.5 баллов

Велосипедист и тренер отправились одновременно с тренировочной базы на стадион. Велосипедист поехал со скоростью **30** км/ч, доехал до стадиона и поехал назад навстречу тренеру, снизив скорость в два раза. Встретились они через **9** минут. Сколько метров проехал велосипедист, если тренер шёл со скоростью **5** км/ч?

Число

№ 14

0.5 баллов

На доске написано число **2023**. Если на доске написано число x , то за один ход можно дописать на доску число $x - 3$ или $2x + 6$. Какие числа, большие **10** и меньше **20**, можно получить с помощью таких операций?

Число



№ 15

0.5 баллов

Дан фрагмент псевдокода на универсальном языке программирования. В языке есть все привычные конструкции: арифметические и логические операции, переменные, встроенные структуры данных, ветвления, циклы, функции.

Так, в языке есть конструкция `dict` — ассоциативный массив (аналогичный словарю из Python). Чтобы достать значение из ассоциативного массива по какому-то ключу, нужно написать `d[k]`, где `d` — переменная, в которой хранится ассоциативный массив, а `k` — ключ, по которому происходит обращение.

Чтобы создать ассоциативный массив с заданным соответствием ключей и значений, нужно прописать содержимое массива в фигурных скобках. Например, запись `{'a': 1, 'b': 2}` подразумевает создание ассоциативного массива, который сопоставляет букве «a» число 1, а букве «b» — число 2.

Также в языке есть строки. Чтобы создать строку, необходимо прописать в одинарных кавычках её содержимое: `s = 'abcd'`. Для обращения к конкретному символу строки используются квадратные скобки с указанием индекса этого символа (считая от 0): `s[2]`. Длину строки можно узнать с помощью функции `len`.

Цикл `for` в языке обозначается как `for i in 1 .. r`, где `i` — индекс текущей итерации цикла, `1` — наименьшее значение, которое принимает `i` в цикле, `r` — максимальное. В ходе работы цикла он пройдёт `r - 1 + 1` итерацию, во время которых `i` примет все значения от `1` до `r` включительно.

В языке проверка равенства двух объектов производится с помощью оператора `==`. Отрицание логического выражения — с помощью конструкции `not`.

Фрагмент кода на языке представлен ниже.

```
d = {'a': 1, 'b': 2, 'c': 3, 'd': 4, '-': -1}
s = 'ab--b-daa---a'

result = 0
prev_minus = False

for i in 0 .. (len(s) - 1):
    letter = s[i]

    if letter == '-':
        if not prev_minus:
            result += d[letter]
            prev_minus = True
        else:
            result += d[letter]
            prev_minus = False
```

В качестве ответа на задание укажите значение, которое будет находиться в переменной `result` после исполнения приведённого фрагмента кода.

Число

№ 16

0.5 баллов

Дан фрагмент псевдокода на универсальном языке программирования. В языке есть все привычные конструкции: арифметические и логические операции, переменные, встроенные структуры данных, ветвления, циклы, функции.

Так, в языке есть конструкция **list** — список (аналогичный списку из Python), к элементам которого можно обращаться по индексу с помощью квадратных скобок и к которому можно присоединять новые элементы с помощью функции **add**.

Чтобы создать пустой список, необходимо прописать пустые квадратные скобки — **[]**. Чтобы создать непустой список, нужно прописать содержимое списка в квадратных скобках: **[1, 2, 3]**.

В языке есть функция **len**, которая возвращает длину списка. Также в языке есть функция **reverse**, которая разворачивает список: например, из списка **[1, 2, 3]** функция сделает список **[3, 2, 1]**.

Цикл **for** в языке обозначается как **for i in l .. r**, где **i** — индекс текущей итерации цикла, **l** — наименьшее значение, которое принимает **i** в цикле, **r** — максимальное. В ходе работы цикла он пройдёт **r - l + 1** итерацию, во время которых **i** примет все значения от **l** до **r** включительно.

В языке проверка равенства двух объектов производится с помощью оператора **==**. Логическое «И» обозначается с помощью оператора **and**.

Фрагмент кода на языке представлен ниже.

```
function f(l1, l2):  
    i = 0  
    j = 0  
  
    res = []  
  
    while (i < len(l1)) and (j < len(l2)):  
        if l1[i] < l2[j]:  
            res.add(l1[i])  
            i += 1  
        else:  
            res.add(l2[j])  
            j += 1  
  
    if i == len(l1):  
        for k in j .. (len(l2) - 1):  
            res.add(l2[k])  
  
    if j == len(l2):  
        for k in i .. (len(l1) - 1):  
            res.add(l1[k])  
  
    return res
```

```
function g(l):  
    l1 = []  
    l2 = []  
  
    for i in 0 .. (len(l) - 1):  
        x = l[i]  
        x_2 = x * x  
  
        if x < 0:  
            l1.add(x_2)  
        else:  
            l2.add(x_2)  
  
    return f(reverse(l1), l2)
```

```
result = g([-9, -7, -3, 0, 0, 2, 3, 10])
```

Какой список будет находится в переменной **result** в результате исполнения приведённого фрагмента кода?

В качестве ответа укажите элементы списка **result** в порядке их следования, добавляя необходимое количество полей ответа. Каждый элемент записывайте в отдельное поле.

Число или дробь



№ 17

1.5 баллов

Вам дан [zip-архив](#) с текстовым файлом. Файл содержит 100 000 целых неотрицательных чисел. Числа записаны в одну строчку и разделены пробелами.

Какое число встречается в файле наиболее часто?

Если таких чисел несколько, укажите их все в любом порядке.

Число



№ 18

2 балла

Вам дан [zip-архив](#). Внутри архива содержится текстовый файл, который представляет из себя матрицу смежности неориентированного графа.

Число строк в файле (обозначим его N) равно числу вершин в графе. Каждая строка соответствует отдельной вершине графа и содержит N чисел, записанных через запятую (,). Если в i -й строке на j -й позиции стоит 1, то это значит, что в графе между вершинами i и j проведено ребро. Если на соответствующей позиции стоит 0, то это значит, что ребро между вершинами отсутствует. Индексация вершин начинается с 0.

Найдите для каждой из следующих пар индексов длину **кратчайшего** пути между соответствующими вершинами:

Индекс первой вершины	Индекс второй вершины	Длина пути
0	44	
1	25	
76	54	
89	99	

2 балла

Вы работаете аналитиком данных крупной социальной сети. В социальной сети у каждого пользователя есть возможность сделать публикацию, а также оценить публикацию другого пользователя, поставив на ней отметку «Отлично».

Вам дан [zip-архив](#), в котором содержатся две таблицы в формате [CSV](#): `post_user.csv` и `post_like.csv`.

В таблице `post_user.csv` содержится информация про связь публикаций и пользователей, которые их сделали. Таблица содержит следующие колонки:

1. `post_id` — идентификатор публикации.
2. `user_id` — идентификатор пользователя, который сделал публикацию.

В таблице `post_like.csv` содержится информация про количество отметок «Отлично», которые получили пользовательские публикации. Таблица содержит следующие колонки:

1. `post_id` — идентификатор публикации.
2. `likes_cnt` — количество отметок «Отлично» на публикации.

Публикации из таблицы `post_user.csv` можно сопоставить с публикациями из таблицы `post_like.csv` с помощью колонки `post_id`: для публикации с заданным `post_id` из таблицы `post_user.csv` можно посмотреть число отметок «Отлично» в той строчке таблицы `post_like.csv`, для которой значение колонки `post_id` равно заданному.

По представленным данным вам необходимо определить пользователя с наибольшим количеством публикаций, набравших не менее 50 отметок «Отлично».

В качестве ответа укажите идентификатор этого пользователя. Если таких пользователей несколько, укажите **наибольший** из их идентификаторов.

Число

2 балла

Модель линейной регрессии — одна из самых простых с точки зрения работы среди моделей, которые используются в машинном обучении. Для данного объекта с характеристиками $x_1, x_2, \dots, x_m$ модель предсказывает значение целевой характеристики по следующей формуле:

$$\hat{y} = w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \dots + w_m \cdot x_m,$$

где $w_0, w_1, w_2, \dots, w_m$ — параметры модели, а m — общее число характеристик, описывающих объект. *Предсказанное* моделью значение обозначают $\hat{y}$.

Вам дана модель линейной регрессии, которая для объекта с параметрами x_1 и x_2 вычисляет значение $\hat{y}$ по следующей формуле:

$$\hat{y} = 2 + 3 \cdot x_1 - 4 \cdot x_2.$$

Также вам дан [zip-архив](#), который содержит таблицу в формате [CSV](#). Каждая строка таблицы соответствует одному объекту. Для объекта известны:

1. Значение параметра x_1 . Оно находится в колонке **x1**.
2. Значение параметра x_2 . Оно находится в колонке **x2**.
3. Значение характеристики y , которую модель линейной регрессии должна уметь предсказывать. Оно находится в колонке **y**.

Для каждого объекта, представленного в файле, с помощью данной модели линейной регрессии рассчитайте значение $\hat{y}$, после чего вычислите **ошибку** предсказания модели по формуле:

$$\frac{(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \dots + (y_n - \hat{y}_n)^2}{n},$$

где y_i — значение характеристики y для i -го объекта, $\hat{y}_i$ — предсказанное моделью значение характеристики y для i -го объекта, а n — число объектов в файле.

В качестве ответа введите полученное значение ошибки. Ответ **округлите до целых вниз**.

Примечания

1. Пример работы модели линейной регрессии.

Допустим, что для объекта значение характеристики x_1 равно 2, а значение характеристики x_2 равно 1. Тогда значение, предсказанное данной в задании моделью линейной регрессии, будет равно:

$$\hat{y} = 2 + 3 \cdot 2 - 4 \cdot 1 = 4.$$

2. Работа с большими суммами.

В формуле для вычисления значения ошибки на месте многоточия должны стоять слагаемые суммы, которые соответствуют объектам с порядковыми номерами 3, 4 и так далее до $n - 1$. Например, если бы в данных было 5 объектов, то формула бы имела следующий вид:

$$\frac{(y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + (y_3 - \hat{y}_3)^2 + (y_4 - \hat{y}_4)^2 + (y_5 - \hat{y}_5)^2}{5}.$$

Число