

2019/20 учебный год

**Всероссийская олимпиада студентов «Я – профессионал»**

**Демонстрационный вариант**

задания заключительного (очного) этапа

по направлению «**Искусственный интеллект**»

Категория участия: «Магистратура/специалитет»

(для поступающих в аспирантуру/ординатуру)

## Часть 1

### Задача 1. OmegaGo

OmegaGo проводит сеанс одновременной игры в Го с  $n$  игроками. OmegaGo обладает определенным объемом вычислительных мощностей  $S$  и должна их распределить между  $n$  партиями, то есть выбрать такие действительные числа  $b_1, b_2, \dots, b_n$ , что  $0 < b_i \leq S$  и  $b_1 + b_2 + \dots + b_n = S$ .

Вероятность, что OmegaGo с вычислительной мощностью  $b$  выиграет участника силы  $a$ , равна  $P(b, a) = e^{-\frac{a}{b}}$ . Ничьих в Го не бывает.

Напишите программу, находящую оптимальный способ распределить вычислительные мощности, если требуется максимизировать вероятность выигрыша во всех партиях.

#### Формат ввода

На первой строке два натуральных числа  $n$  и  $S$ , разделенных пробелом.

$1 \leq n \leq 100$ ,  $1 \leq S \leq 100000$ .

На второй строке  $n$  натуральных чисел, разделенных пробелами:  $a_1, a_2, \dots, a_n$  — силы игроков.  $1 \leq a_i \leq 10000$ .

#### Формат вывода

В единственной строке выведите  $n$  действительных чисел, разделенных пробелами  $b_1, b_2, \dots, b_n$ .  $b_i$  вычислительная мощность, которую OmegaGo выделяет на  $i$ -того игрока в оптимальном случае ( $0 < b_i \leq S$ ). Ответ будет засчитан, если каждое число в ответе будет отличаться от его истинного значения не более чем на 0.01.

#### Пример 1

Ввод	Вывод
2 10 1 1	5.0 5.0

#### Пример 2

Ввод	Вывод
3 3000 4354 3554 2453	1130.3417078645105 1021.2316172773144 848.4266748581751

#### Тесты и решение

[https://yadi.sk/d/RPW\\_DjHWFGGoBBQ](https://yadi.sk/d/RPW_DjHWFGGoBBQ)

### Задача 2. GuessNumber

Загадано натуральное число  $x$  от 1 до  $10^5$ . Вам нужно написать программу, отгадывающую это число. Программа может задавать вопросы вида «Верно ли, что  $x < a$ » для различных  $a$ , но ответы на эти вопросы она получает не сразу. Вопросы задаются раундами — в одном раунде можно задать

произвольное количество вопросов и по окончании раунда получить ответы на все эти вопросы. За каждый вопрос нужно заплатить 1 монету, за каждый раунд 10 монет. Нужно привести алгоритм, который отгадает число  $x$ , заплатив не больше чем 100 монет.

Не забывайте делать операцию 'flush' после вывода ответа. За вывод ответа платить не нужно.

Вы получите вердикт Wrong Answer, если:

- Вы потратили строго больше 100 монет.
- Итоговый ответ не является правильным.

Вы получите вердикт Presentation Error, если:

- Вопрос или ответ не соответствует описанному выше формату.
- Параметр  $a$  в вашем запросе меньше 1 или больше 100000.

Вы получите вердикт Idleness Limit Exceeded, если не будете ничего выводить (а тестирующая программа будет ожидать ввода) или забудете сделать операцию 'flush' после какого-нибудь вывода (смотрите ниже).

Если ваша программа, написанная на интерпретируемом языке (например, Python), получает какую-либо ошибку исполнения, во время общения с интерактором, то возможно получение любого вердикта Presentation Error/Wrong Answer/Idleness Limit Exceeded, поскольку сообщение об ошибке передается интерактору и может быть воспринято по-разному.

### Формат ввода

Это интерактивная задача.

Тестируемой программе не подается никаких входных данных.

Тестируемая программа пишет запросы и получает на них ответы от программы интерактора.

### Формат вывода

Ваша программа может отправлять следующие запросы:

- «?  $a$ » — запрос «Верно ли, что  $x < a$ », где  $a$  натуральное число от 1 до 100000. Стоимость одна монета.
- «+» — конец раунда. Стоимость 10 монет. В ответ на этот запрос приходят ответы на все запросы вида «?  $a$ », заданные в этом раунде. Если  $x < a$  придет ответ 1, если нет, то ответ будет 0. Каждый ответ приходит на отдельной строке в том порядке, в каком вопросы были заданы.
- «!  $x$ » — угадывание числа  $x$ . Сразу после этого ваша программа должна завершиться.

Каждый запрос должен быть на отдельной строке. Вывод, не совпадающий ни с одним запросом, будет приводить к ошибке представления.

Для сброса буфера вывода (то есть для операции 'flush') сразу после вывода запроса можно сделать:

- `fflush(stdout)` в языке C++;
- `System.out.flush()` в Java;
- `stdout.flush()` в Python;
- `flush(output)` в Pascal; смотрите документацию для других языков.

## Примечания

Приведем пример корректного взаимодействия программы и интерактора, если загадано число 5.

Строчки с выводом интерактора отмечены знаками »

```
? 2
? 4
? 6
+
»0
»0
»1
? 5
+
»0
! 5
```

## Тесты и решение

[https://yadi.sk/d/m\\_4UqT1cST7jZg](https://yadi.sk/d/m_4UqT1cST7jZg)

## Задача 3. Cryptocurrency

Вася работает аналитиком криптовалютного рынка.

Каждый день он должен предсказывать на завтра курс какой-нибудь криптовалюты. За любое несоответствие прогноза реальной ситуации Васю штрафуют величиной  $|p_{\text{real}} - p_{\text{predict}}|$ , где  $p_{\text{predict}}$  — предсказанная Васей стоимость валюты, а  $p_{\text{real}}$  — настоящая стоимость.

Поскольку Вася работает аналитиком уже давно, он понимает, что  $p_{\text{real}}(X, t)$  — цена криптовалюты  $X$  в день  $t$  — является случайной величиной с распределением  $D_X$ , которое зависит только от названия криптовалюты. Более того, курсы разных криптовалют независимы в совокупности, и даже курс криптовалюты  $X$  в день  $t$  не зависит от ее курса во все предыдущие дни.

К предсказанию доступны курсы криптовалют  $A, B, C$ . Перед первым предсказанием Вася узнал их цены за последние 3 дня.

Напишите программу, которая позволит Васе сделать так, чтобы штраф за следующие  $n$  дней был меньше, чем его зарплата  $S$ .

Задача интерактивная. Чтобы предсказать курс, ваша программа должна вывести  $X$  price, где  $X$  это  $A, B$  или  $C$ , а price — предсказываемая цена. В ответ она получит строчку из 3 чисел, разделенных пробелами — реальные курсы валют  $A, B, C$  за этот день. Ваша программа должна произвести  $n$  предсказаний.

Не забывайте делать операцию 'flush' после вывода ответа.

Вы получите вердикт Wrong Answer, если суммарный штраф за  $n$  дней оказался больше или равен чем зарплата Васи  $S$ .

Вы получите вердикт Presentation Error, если ваш запрос не соответствует описанному выше формату.

Вы получите вердикт Idleness Limit Exceeded, если не будете ничего выводить (а тестирующая программа будет ожидать ввода) или забудете сделать операцию 'flush' после какого-нибудь вывода (смотрите ниже).

Если ваша программа, написанная на интерпретируемом языке (например, Python), получает какую-либо ошибку исполнения, во время общения с интерактором, то возможно получение любого вердикта

Presentation Error/Wrong Answer/Idleness Limit Exceeded, поскольку сообщение об ошибке передается интерактору и может быть воспринято по-разному.

### Формат ввода

Вначале программа получает 4 строки.

В первой строке идут два целых числа, разделенных пробелом,  $n$  — количество дней и  $S$  — зарплата Васи.  $1 \leq n \leq 100$ ,  $1 \leq S \leq 10^5$

Далее идут 3 строки, каждая содержит 3 действительных числа, разделенных пробелом, —  $a, b, c$  — цены соответствующих криптовалют за последние 3 дня.

На каждую команду вида  $X \text{ price}$  придет ответ:

3 действительных числа, разделенных пробелом, —  $a, b, c$  — реальные цены криптовалют.

### Формат вывода

Ваша программа должна предсказывать курс криптовалюты с помощью запросов вида  $X \text{ price}$ , где  $X$  — название криптовалюты (A, B или C), и  $\text{price}$  — действительное число.

На каждую команду  $X \text{ price}$  придет ответ: 3 действительных числа, разделенных пробелом, —  $a, b, c$  — реальные цены криптовалют в текущий день.

Сразу после вывода последнего предсказания ваша программа должна завершиться. Каждый запрос должен быть на отдельной строке. Вывод, не совпадающий ни с одним запросом, будет приводить к ошибке представления.

Для сброса буфера вывода (то есть для операции 'flush') сразу после вывода запроса можно сделать:

- `fflush(stdout)` в языке C++;
- `System.out.flush()` в Java;
- `stdout.flush()` в Python;
- `flush(output)` в Pascal; смотрите документацию для других языков.

### Примечания

Приведем пример корректной работы программы. Программа предсказывает случайно выбранную криптовалюту значением. В качестве предсказания цены в день  $t$  берется значение цены в день  $t-1$ . Вывод интерактора = вход вашей программы обозначен символами ».

Вначале программа получает:

```
»3 5
»1.0 4.0 6.0
»2.0 3.0 6.0
»1.0 4.0 5.0
```

Программа должна сделать  $n=3$  предсказаний и получить штраф не больше чем  $S=5$ .

В первый день программа предсказывает курс криптовалюты A значением 1:

```
A 1
»2.0 4.0 5.0
```

Вася получает штраф  $|1-2|=1$ .

Во второй день программа предсказывает курс криптовалюты C значением 5:

```
C 5
»2.0 3.0 5.0
```

Вася получает штраф  $|5-5|=0$ .

В третий день программа предсказывает курс криптовалюты А значением 2:

A 2

»1.0 4.0 6.0

Вся получает штраф  $|2-2|=1$ .

После этого программа завершается. Суммарный штраф  $1+0+1=2<5=S$ . Программа прошла данный тест.

### Тесты и решение

<https://yadi.sk/d/uiTcAH7dt4hdvQ>

### Задача 4. Maze

Лабиринт представляет собой клетчатый квадрат 10 на 10, в некоторых клетках стоят препятствия, а в одной клетке находится выход. Робот находится в таком лабиринте и может выполнять 4 команды: сдвинуться на одну клетку вниз, вверх, вправо или влево. Если робот пытается выйти за границы лабиринта или перейти в клетку с препятствием, то он остается на месте. Если робот попадает в выход, то он выходит из лабиринта и дальнейшие команды игнорирует. Напишите программу для робота, исполняя которую робот в любом случае доберется до выхода вне зависимости от клетки, в которой он находился вначале.

Программа должна состоять из не более чем 1000 команд.

Лабиринт:

```
□□■□■□□■□□
□■□□□□■□□□
□■□□X□□□□□
□□■□□□□■□□
□□□□■□□□□□
□□□□■□□■□□
□□□□□■□□■□
□■□□■□■□■□
□□■□□□■□■□
■□□□□■□□□□
```

### Формат ввода

Ввода нет. Нужно написать программу для одного конкретного указанного в условии лабиринта.

Версия лабиринта, которую можно скопировать. 0 — свободная клетка, 1 — препятствие, x — выход.

0011010011

0100001000

0110x00000

0010000100

0000111000

0000100100

0000010010

0100101010

0011001010

1000011000

## Формат вывода

Одна строка, состоящая из символов U,D,R,L длины не более 1000.

## Подходящий ответ

DLDLRDDDDDDLLLRDDDDRRUUUUURLLDLLLULUURRRUUUURDLLLLL

## Задача 5. Polygon

Вася изучает два способа генерации выпуклых многоугольников.

Первый: выбирается 50 случайных точек с действительными координатами в квадрате  $[0,100] \times [0,100]$  и рассматривается их выпуклая оболочка. Случайная точка в квадрате выбирается так: из равномерного на отрезке  $[0,100]$  распределения независимо выбираются координаты  $x$  и  $y$ .

Второй: Выбирается случайное число  $m$  от 2 до 19 и генерируются  $m$  случайных векторов следующим образом: действительные числа  $u$  и  $v$  выбираются независимо из равномерного на отрезке  $[-50,50]$  распределения и рассматривается вектор  $(u,v)$ . Затем добавляется  $m+1$ -ый вектор такой, что сумма всех векторов становится равной нулевому вектору, и все вектора сортируются по возрастанию направленного угла с осью  $X$  и в таком порядке рассматриваются как стороны выпуклого многоугольника.

Выборка по 5000 многоугольников сгенерированных этими способами:

<https://yadi.sk/d/CAIN2ZEtiBwhbA>

Вася сгенерировал по 50 многоугольников обоих типов, отцентровал их — сдвинул координаты так, что центр масс вершин многоугольников попал в точку  $(0,0)$ , и напечатал координаты вершин на карточках. Из-за неловкости Васи все карточки перепутались. Помогите Васе определить, каким образом был сгенерирован каждый многоугольник.

В каждом тесте можно ошибиться не более чем для 5 многоугольников из 100.

## Формат ввода

В первой строке каждого теста находится  $N=100$  количество многоугольников в данном тесте.

Затем заданы  $N$  многоугольников в следующем формате. На отдельной строке написано  $M_i$  — количество вершин в  $i$ -том многоугольнике, затем следует перечисление вершин этого многоугольника — каждая вершина задана на своей строке парой действительных чисел, разделенных пробелом, в начале  $x$ -координата, потом  $y$ -координата. Вершины многоугольников даны в порядке обхода.

## Формат вывода

Для каждого теста на отдельной строке выведите Points, если вы считаете, что многоугольник был сгенерирован случайным выбором точек или Sides, если считаете, что многоугольник был сгенерирован случайным выбором сторон.

## Примечания

В каждом тесте 100 многоугольников. Нужно верно определить, как были сгенерированы хотя бы 95 из них.

## Тесты и решение

<https://yadi.sk/d/fDdamBzuQbnmQw>

## Часть 2

### Задача 1. Lunch

Задача от индустриального партнера «Тинькофф Банк».

В Тинькофф банке для сотрудников действует бесплатная столовая – Тинькофф кафе. Работники имеют возможность выбрать любые блюда, которые им нравятся и имеются в наличии. Задача: разработать алгоритм, который сможет определить сотрудника по его предпочтениям в еде.

Метрикой качества является ассигасу — доля верных предсказаний.

Данные: <https://yadi.sk/d/TAR7mlQYd4Hs1A>

### Формат ввода

Данные представляют собой запись количества взятых на обед блюд сотрудниками Тинькофф за определенный промежуток времени. Колонки `good_0`, `good_1`, `good_2` и так далее в файлах `train`, `eval` и `test` — это разные типы блюд, их расшифровка доступна в файле `goods.csv`. Каждая строка в `train`, `eval` и `test` — это история обедов уникального сотрудника, идентификатор которого доступен в колонке `target`.

Описание данных:

- `train.csv` — данные для обучения
- `eval.csv` — данные для валидации
- `test.csv` — данные, которые нужно предсказать
- `goods.csv` — сопоставление реальных блюд и их идентификаторов

### Формат вывода

Требуется предсказать колонку `target1` для файла `test.csv` и отправить предсказания в тестирующую систему.

Файл с ответом должен содержать 3021 строку: Первая строка должна быть

"ID,target"

остальные 3020 строк содержат предсказания вида "id,target\_x", где `id` номер строки, начиная с 0, а `x` — предсказанный человек. Пробелов или других символов после запятой или в конце строки быть не должно.

ID,target

0,person\_0

1,person\_1

2,person\_2

...

...

3019,person\_3019

- `sample_sub.csv` — пример сабмита

### Ответ

<https://yadi.sk/i/coFN9eQ7jUo-AA>



## Задача 2. TextRecognition

Задача от индустриального партнера «IC».

Даны изображения с текстом на английском языке, которые испорчены вертикальными белыми полосами. Все картинки можно разделить на 3 типа:

- Сгенерированный осмысленный текст;
- Осмысленный текст, взятый из реальных книг;
- Бессмысленный текст (буквы, цифры и пробелы).

Имеется обучающая выборка из 700 картинок и тестовая из 100.

Задача: распознать текст на картинках из тестовой выборки.

$$\frac{LD(text_{real}, text_{predicted})}{|text_{real}|}$$

Для каждого изображения ошибка вычисляется как  $\frac{LD(text_{real}, text_{predicted})}{|text_{real}|}$ . Где  $LD(text_{real}, text_{predicted})$  — расстояние Левенштейна между предсказанным текстом и реальным текстом на картинке. Расстояние Левенштейна между двумя текстами — это минимальное количество операций вставки, удаления или замены одного символа, с помощью которых один текст можно превратить в другой. Пробельные символы, несколько пробельных символов подряд, переводы строки считаются одним и тем же вспомогательным символом. Общая ошибка есть сумма ошибок по изображениям из тестовой выборки.

Данные: <https://yadi.sk/d/tNatiXXiGOjDig>

### Формат ввода

Обучающая выборка: 700 файлов result\_i.png и result\_i.txt, где  $0 \leq i \leq 699$ . Тестовая выборка: 100 файлов result\_i.png, где  $700 \leq i \leq 799$ .

### Формат вывода

В качестве ответа принимается один файл. Файл должен состоять из 100 блоков, в следующем формате.

```
700 {
Распознанный_текст_из_файла_result_700.png
}
701 {
Распознанный_текст_из_файла_result_701.png
}
...
...
799 {
Распознанный_текст_из_файла_result_799.png
}
```

Текст каждого блока должен занимать не более 1000 строк. Блок заканчивается строкой из единственного символа "}".

### Ответ

<https://yadi.sk/d/7kEJPILLV72mHw>

### **Задача 3. Catalog**

Задача от индустриального партнера «ТМК».

Есть справочник различных позиций, используемых компанией ТМК. Каждая позиция в справочнике содержит всего два атрибута: "Название" и "Группа". Задача: предсказывать атрибут "Группа" по атрибуту "Название".

Метрикой качества является ассигасу — доля верных предсказаний.

Данные: <https://yadi.sk/d/prbZ9YkK17q2QA>

#### **Формат ввода**

train.txt — файл с обучающей выборкой: каждая строка представляет собой одну позицию и состоит из названия позиции и группы, разделенных символом табуляции.

test.txt — файл с тестовой выборкой: файл состоит из 2346 строк, каждая строка полностью состоит из названия позиции, для которого нужно определить группу.

#### **Формат вывода**

Ответ требуется в следующем формате: файл из 2346 строк, *i*-ая строка должно представлять собой предсказанную группу для *i*-ой строки из файла test.txt.

Группы сравниваются посимвольно, не должно быть лишних пробельных и любых других символов.

#### **Ответ**

<https://yadi.sk/i/eGc9SxUfEWyjtA>