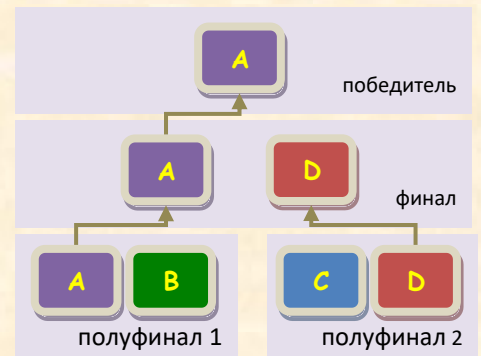


Versus → speedCoding

Вы наверняка знаете кто такой Ресторатор и что такое Versus Battle (лат. versus — против; англ. battle — битва). Это интересно, модно, привлекательно, даже опасно, но мы не будем опускаться до обценной лексики. Мы будем использовать лексику C#, Python, Pascal, C++, JavaScript - мы будем проводить **поединки** между программистами.

Отсев будет происходить по **Олимпийской системе**, то есть тот, кто в поединке проигрывает, тот выбывает, а победитель идёт на битву с победителем другой пары. Смотрите пример из схемы.



В чём состоит поединок и как он проходит?

Два этапа по 5 заданий.

Первый этап: это вопросы на понимание синтаксиса языка (C#, JS, Python).

Пример: что выведет в консоль данная программа:

```
var g = 0;
g = 1 && g++;
console.log(g);
```

Каждый участник записывает свой вариант ответа в консоль, после чего ответы обоих участников сравниваются с эталонным. За каждый правильный ответ +1 балл.

Второй этап: это решение задач, за каждое решение +1 балл.

Два участника одновременно получают простую задачу. Выполняют её, кто первый получает правильный результат, поднимает руку и провозглашает «**Раунд**». Язык программирования каждый выбирает сам.

Писать код нужно будет **публично** - ваш ноутбук должен быть подключен к проектору и отображать все ваши действия. У каждого участника ноутбук и проектор. Можно пользоваться любыми источниками.

К задаче прилагается вариант входного потока данных и правильного ответа для контроля работоспособности программы во время редактирования. И ещё один вариант входного потока без ответа - для итогового контроля решения путём сверки с эталонным. Если участник сказал «Раунд», но его решение не совпало с эталонным, то второму участнику даётся минута для завершения решения. Как правило, входные данные будут поступать из файла с именем input.txt, а вывод результата нужно будет делать на экран / в консоль. Поэтому участникам поединка разрешено заранее создать пустой файл input.txt и набрать заготовку программы, например, для языка Python:

```
lines = open('input.txt', 'r').readlines()
```

```
print()
```

Одну и ту же заготовку можно будет использовать для всех последующих заданий, но не допускается делать отдельные заготовки на все задачи.

Задачи будут в рамках уровня познаний первого курса: циклы, ветвления, строки, списки, арифметика и логика с небольшой изюминкой. Сложность будет подбираться так, чтобы задачу можно было бы выполнить за 2-5 минут и уложиться в 5-15 строчек кода (см. примеры в приложении). Всего 5 задач на поединок, следовательно, можно будет уложиться в 30 минут (а битва из трёх поединков - 1,5 часа). Так как акцент в битве не на решении сложных задач, а на скорости выполнения, то название мероприятия *Versus speedCoding*, а участники спидкодеры. За полтора часа можно провести battle на четырех кодеров.

Такие поединки можно проводить регулярно, за игроками будет вестись рейтинг. Он будет влиять на авторитет кодера и учитываться при следующих встречах при посеве участников, чтобы сильные игроки не попали сразу друг на друга на этапе полуфинала.

Победитель получает уважение и почёт, силиконовый браслет на руку и линк на сертификат о победе (для подтверждения профессиональных навыков и просто для пафоса). Если будут спонсоры, то они сами решат, что подарить...

Нужны волонтёры:

- Ведущий (язык «подвешен»),
- Техник (проекторы, ноутбуки и среды программирования),
- Репортёр (запись фото, видео, аудио).

Ну что, ПОШУМИМ?

Приложение

Task_0

Сеньор Помидор в текстовом файле хранит номер своего счета. Номер счёта это просто последовательность цифр, находящихся среди текста. Мистер Овальный хочет узнать номер счёта, помогите ему программой.

Пример.

Вход:

Сеньор помидор4 в 02текстовом файле хранит но0мер своего счета. Номер счёта это просто последовательность цифббр, находящихся среди текста. Мистер 09в7а5льный хочет узнать номер счёта, помогите ему программой.

Выход:

402066975

Task_1

В текстовом файле символами `` ` `` написан индекс отправителя письма. Остальные символы пробелы. Нужно подсчитать, сколько символов истрачено.

Вход:

```
* ****
**   *
* *   *
*   *
*   *
* ****
```

Выход:

20