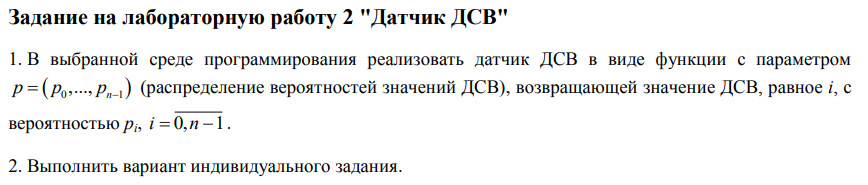
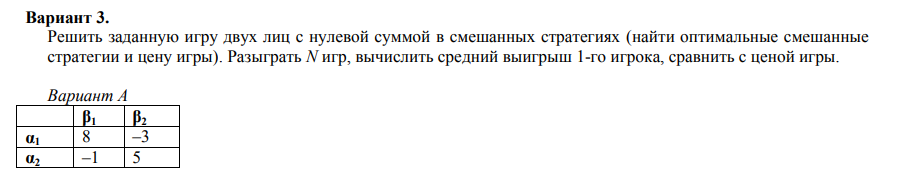
# Имитационное моделирование

# Отчет по 2 лабораторной работе

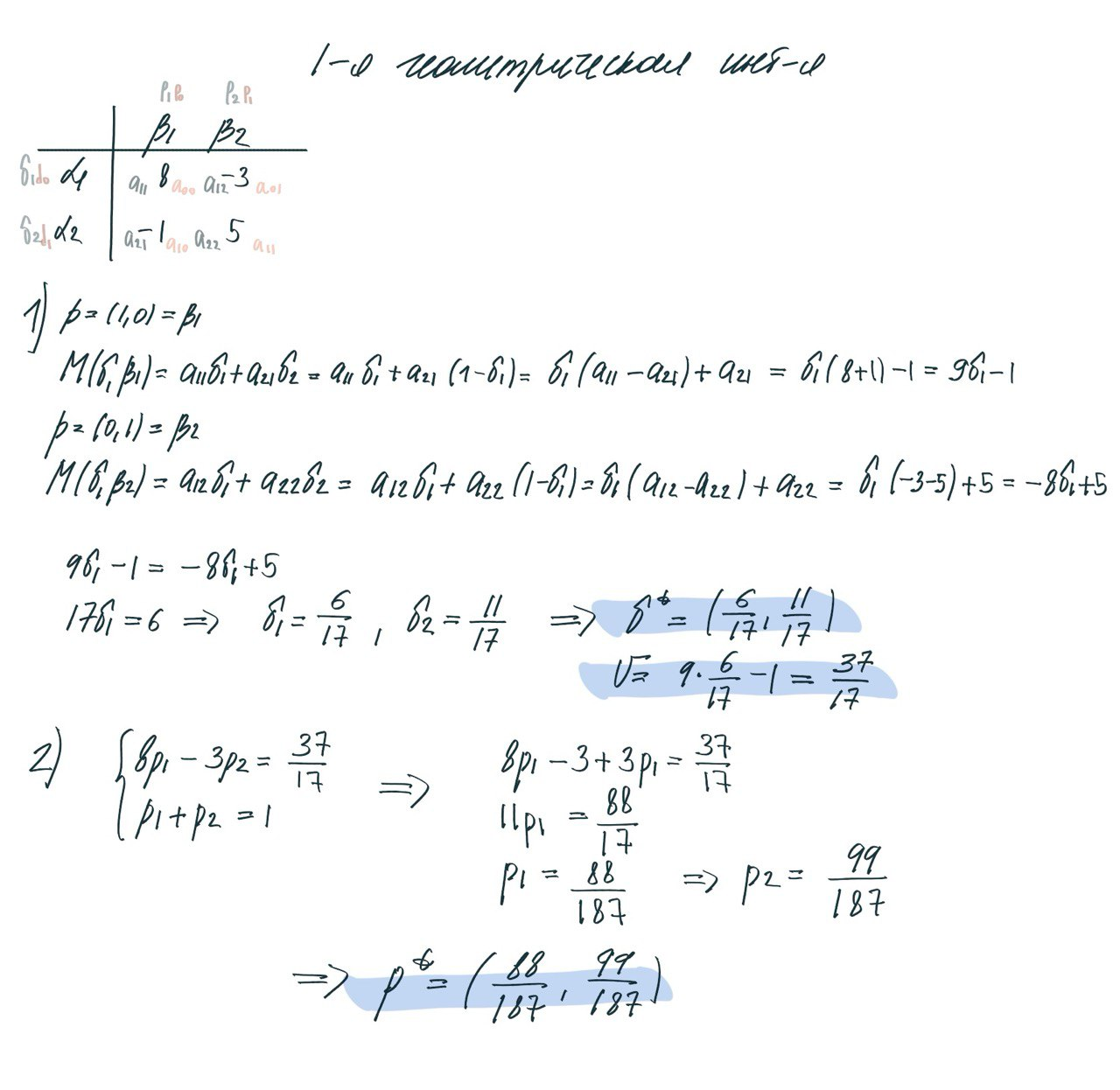




При помощи random.uniform(0.0,1.0) создаем функцию BRV (basic random variable), которая будет возвращать случайную величину, равномерно распределенную на интервале (0,1). Причем, если создастся число =0 или =1, то функция вызовет сама себя и пересоздаст его. Так мы гарантируем, что в нашей выборке не будет чисел 0 и 1

В функцию DRV (discrete random variable) передаем массив вероятностей. Далее генерируем значение из ф-ии BRV и заводим счетчик, который мы и будем возвращать как итог выполнения функции. Из сгенерированной БСВ мы вычитаем вероятность, переданную в функцию, и определяем в каком интервале длины probability[i] находится БСВ.   
Если БСВ все еще больше 0, то значит это не тот интервал, который нам нужен, а, следовательно, повторяем вычисления

Функция First\_Geometric\_Interpretation(a) решает игру двух лиц с нулевой суммой в смешанных стратегиях (+ ищет цену игры) методом 1-ой геометрической интерпретации. Ниже «ручная» проверка значений вероятностей и формулы, которые я использовала в реализованной функции.



Подключаем библиотеку sympy и встроенные в нее функции symbols (создает символьные переменные), solve (решает уравнение), Eq (составляет его). С их помощью ищем значения вероятностей d1 и p1 для 1 и 2 игроков, а также вычисляем цену игры v. В конце своей работы функция возвращает v, и две пары вероятностей для первого и второго игрока.

Функция First\_Player\_Wins(strategy1, strategy2, a, N) на вход получает пару вероятностей для 1 игрока, пару вероятностей для 2 игрока, изначальный массив значений и количество проводимых испытаний.

Дважды обращаясь к функции, реализующей ДСВ (которая формирует БСВ и возвращает номер интервала), мы вычисляем результат конкретной партии и складываем его с общим числом выигрыша первого игрока. Возвращаем результат деления общей суммы «побед» на количество проведенных партий