A. *Задача об очках*

Изучение азартных случайных игр восходит к ранней истории теории вероятностей, и даже сейчас эти игры доставляют множество тренировочных задач как для начинающих, так и для специалистов.

Около 1654 г. любитель математики шевалье де Мере консультировался с Блезом Паскалем (знаменитым математиком, естествоиспытателем и философом) относительно решения некоторых обобщений следующей задачи.

Двое игроков играют серию партий, причём в каждой партии какой-то из двух игроков выигрывает одно очко; шансы выигрыша для этих игроков каждый раз равны. Тот из игроков, который первым наберет три очка, забирает обе ставки. Игроки вынуждены прекратить игру в момент, когда первый игрок набрал 2 очка, а второй — 1 очко.

Как следует им разделить между собой ставки? Паскаль считал, что ставки следует разделить в отношении 3 к 1 в пользу первого игрока.

Напишите программу, моделирующую эту игру. Сколько из этих игр выиграл игрок, уже набравший 2 очка, а сколько — игрок, набравший 1 очко?

B. *Задача о длительности игры*

Два игрока обладают первоначальными капиталами соответственно в M и N единиц. Чтобы участвовать в одной партии игры, каждый должен поставить на кон одну единицу своего капитала. Шансы выигрыша для каждого игрока одинаковы, и выигрывающий партию забирает обе ставки.

Если игра продолжается до разорения какого-либо игрока, то из скольких партий она будет состоять и каковы шансы выигрыша у игрока, начинающего игру с капиталом в M единиц? Рассмотрите задачу о длительности игры при $M = 3, N = 2$:

- Найдите среднее число партий в одной игре;
- Вычислите, какую часть составляли игры, выигранные игроком с начальным капиталом в M единиц.

Смоделируйте задачу для разных M и N:

- Вероятность выигрыша игрока с начальным капиталом в M единиц;
- Вероятность того, что игра будет состоять из T партий.

C. Имеется колода из 52 карт. Карты достаются сверху по одной. Задачи (предполагающие как компьютерное моделирование, так и точные вычисления, возможно, также с помощью компьютера):

- Какова вероятность того, что первый туз появится на 1-м месте? на 2-м месте? …на 52-м месте?
- На каком месте в среднем появляется первый (второй, третий, четвёртый) туз? Вычислите математическое ожидание количества карт, которые надо вытащить из колоды до первого (второго, третьего, четвёртого) туза, двумя способами — моделируя этот процесс и по определению математического ожидания, посчитав соответствующие вероятности.
- Среди какого количества карт, лежащих сверху колоды, в половине всех случаев заведомо находится первый туз?

D. *Количество совпадений*

Выберите из обычной колоды игральных карт (52 карты) две масти — пики и черви. Расположите карты червовой масти в следующем порядке; туз, $2, 3, 4, \ldots, 10$, валет, дама, король.

Перетасуйте карты пиковой масти и выложите их рядом с червями.

Сосчитайте количество совпадений значений карт в этих мастях. Выполните этот эксперимент несколько раз и найдите среднее число совпадений карт.

Попробуйте угадать теоретическое среднее число совпадений.

E. *Случайное блуждание*

В точке 0 числовой оси находится некоторая частица. Каждую секунду она с равной вероятностью сдвигается на 1 либо вправо, либо влево. Вопросы:

- Как далеко в среднем окажется она от точки 0?
- Сколько времени в среднем она будет находиться на положительной полуоси? На отрицательной полуоси?
- Сколько раз в среднем в течение блуждания она будет попадать в точку 0?

F. Парадокс Монти Холла

Имеется три коробки, в одной из которой шарик, а две другие пусты. Есть игрок, цель которого — угадать коробку с шариком и ведущий, который знает, в какой коробке находится шарик.

В начале игрок указывает ведущему на одну из коробок. Ведущий, ничего не говоря о результате выбора игрока, показывает, что в одной из оставшихся коробок шарика нет.

Вопрос: нужно ли игроку менять свой выбор и показать на оставшуюся третью коробку? Увеличит ли смена выбора вероятность угадать коробку с шариком?

Напишите программу, моделирующую игру и выясните стоит ли игроку менять выбранную коробку или нет. Вычислите вероятность выигрыша в обоих случаях.

G. *Теория очередей*

Люди, подходящие случайным образом в различное время к прилавку магазина, за которым их обслуживают, обычно выстраиваются в очередь, если их достаточно много. Если известны данные, касающиеся частоты появления новых покупателей и времени обслуживания одного покупателя, то можно сформулировать, например, такие вопросы:

- Какое время затратит в среднем один покупатель на стояние в очереди?
- В течение какого времени очередь будет состоять более чем из 10 человек?
- Какой эффект даст добавление ещё одного продавца?
- Если людям запрещается становиться в очередь и им приходится покидать магазин, то какой процент покупателей останется необслуженным?

Различные видоизменения этой задачи представляют интерес при исследовании эксплуатационных характеристик комплекса станков, при решении вопроса о количестве контрольных автоматов, которые следует установить на станции метро, при проектировании оборудования, необходимого для телефонных линий или компьютерных сетей, и даже при конструировании и контроле плотин.

H. *Теория эпидемий*

Предположим, что некоторая инфекционная болезнь передается при контакте с заразным больным, причём для предрасположенного к этой болезни человека всегда существует некий шанс заразиться при любом таком контакте, но, раз переболев, он становится невосприимчивым к этой инфекции и не может ее передавать.

Математическая теория эпидемий описывает распространение эпидемий в терминах количества предрасположенных к болезни, заразных и невосприимчивых к инфекции людей в зависимости от времени.

Здесь возникают следующие типичные вопросы:

- Сколько предрасположенных к болезни людей останется после того, как эпидемия кончится, т.е. количество больных станет равным нулю?
- Как долго продлится эпидемия?
- Какова вероятность конца эпидемии для города с заданным количеством жителей?

I. *Функция shuffle*

В языке `Python` есть встроенная функция `shuffle`. Она «перемешивает» элементы массива, являющегося её параметром.

Реализуйте свою функцию `shuffle`, которая генерирует случайную подстановку из чисел $0, 1, \ldots, N-1$, используя только функцию `randint` из модуля `random`.

Реализация должна иметь сложность $O(N)$ в предположении, что один вызов `randint` работает за время $O(1)$.