

Задание №1

Задача:

Создать приложение по переводу значения температуры между градусами цельсия, фаренгейта и кельвина.

Входящие параметры:

Значение температуры вместе с указанием шкалы (цельсий, фаренгейт или кельвин), например: 26C, 299K, или 79F.

Выходные данные:

Целые значения температур в JSON формате для всех шкал измерения, кроме указанной во входящих параметрах, например: {K: 299, F: 79F}

Задание №2

Задача:

Вам необходимо создать приложение для автоматизации работы погрузочных кранов, позволяющих перемещать плиты между грузовыми автомобилями на крупной строительной площадке.

Приложение должно предоставлять кранам последовательность разгрузки, оптимальную для переноса груза с автомобиля на автомобиль, учитывая следующее:

- плиты размещаются на автомобилях одна над другой (от 3 до 8 штук в высоту), и отсортированы по весу (от тяжелых - снизу, к легким - сверху)
- кран может снимать и перемещать только самую верхнюю плиту с грузовика, и при разгрузке не может устанавливать более тяжелые плиты на более легкие
- перенос груза с загруженного автомобиля на пустой нужно осуществить используя только одно дополнительное место для временного хранения плит

Входящие параметры:

Количество плит на автомобиле, который требуется разгрузить (от 3 до 8)



Выходные данные:

Стратегия перемещения плит с автомобиля на автомобиль:

- #1 slot_a -> slot_c
- #2 slot_a -> slot_b
- #1 slot_c -> slot_b
- ...

где: *slot_a* - разгружаемый автомобиль, *slot_b* - дополнительное место для промежуточного хранения плит, *slot c* - пустой автомобиль, куда нужно перенести весь груз, *#n* - номер перемещаемой плиты.

Задание №3

Задача:

Вы участвуете в рыцарском поединке по стрельбе из лука. Поединок начинается с того, что оба соперника располагаются на расстоянии двадцати шагов друг от друга, и каждому дается только одна стрела и, соответственно, только один шанс на выстрел. В течение поединка соперники ходят по очереди, каждый ход участник может либо сделать шаг вперед, либо произвести выстрел. Чем ближе находятся соперники друг к другу, тем выше шанс попадания. При этом шанс попадания увеличивается линейно, начиная с определенного значения в начале поединка до 100% при полном сближении.

Сложность состоит в том, что если вы сделаете выстрел и промахнетесь, соперник будет иметь право сделать необходимое количество шагов, подойти к вам вплотную, и выстрелить с гарантированным попаданием, что обеспечит ему победу.

Создайте алгоритм, который, отталкиваясь от заданных значений шансов попадания для вас и соперника в начале поединка, будет определять номер шага когда вам нужно произвести выстрел для получения наиболее высоких шансов на победу.

Входящие параметры:

Значения шансов попадания в начале поединка (разные для вас и соперника, и оба в диапазоне от 0.1 до 0.3), а также номер участника, который ходит первым.

Выходные данные:

Наиболее оптимальный шаг для произведения выстрела (в диапазоне от 1 до 10).



Задание №4 *

Задача:

Вы работаете с компанией по доставке товаров, которая ежедневно пользуется платной автомобильной дорогой. Плата за путешествие взимается на 10-и пунктах оплаты расположенных вдоль дороги. Водителям компании необходимо преодолеть весь путь, оплатив комиссию за проезд на каждом из пунктов.

Сложность состоит в том, что по правилам, комиссию можно оплачивать только одной единственной монетой. В случае, если ее номинал выше, чем стоимость проезда, водитель сдачу не получает и остаток сгорает. Если же монета, наоборот, не полностью покрывает стоимость проезда, то вашей компании насчитывается долг. При этом стоимость проезда на каждом из пунктов абсолютно произвольно изменяется в конце дня, и может варьироваться в диапазоне от 1-ой до 10-и копеек включительно. Также известно, что несколько пунктов оплаты могут выставлять одну и ту же стоимость проезда, а общая сумма проезда через все пункты будет всегда больше 55-и копеек.

Каждому водителю в начале пути выдается 10 монет, по одной монете каждого достоинства (т.е. одна монета достоинством в копейку, одна монета достоинством в две копейки, одна - три, и так далее, до десяти копеек включительно). Используя генетический алгоритм, вам необходимо найти такую стратегию оплат путешествия, при которой долг водителя в конце пути будет минимальным. Алгоритм будет применяться компанией в начале каждого дня, и использовать данные по новым, только что установленным, размерам комиссий на пунктах оплат для получения новой стратегии для водителей.

Входящие параметры:

Массив из десяти произвольных чисел от 1 до 10, представляющих собой размеры комиссий на каждом из пунктов. Числа в массиве могут повторяться, и их сумма будет всегда больше чем 55.

Выходные данные:

Массив из десяти чисел, представляющих собой достоинства монет, расположенные в порядке, оптимальном для оплат на каждом из пунктов (так чтобы долг компании после всех оплат был минимальным).



Примечания к выполнению заданий

Для задач 2-4 следует дополнительно добавить описание выбранного вами алгоритма, а также, коротко, ваши рассуждения, обосновывающие решение.

Также, во время написания программ, обратите внимание на следующее:

- код приложений должен строго соответствовать методологии ООП, а также быть компактным и легко читаемым
- приложения должны корректно реагировать на широкий спектр возможных входных значений, обрабатывать исключительные ситуации
- все задачи должны быть решены наиболее оптимальным образом, с наименьшим использованием ресурсов памяти и процессора

Выполненное задание (исходный код) присылайте на email jobs@sysgears.com, в качестве темы письма укажите: "Выполненные задания. [Имя Фамилия]".

Дополнительно к письму необходимо прикрепить резюме.