**Операции с массивами**

1. Задан массив, состоящий из N векторов размерности M. Мера схожести векторов X и Y определяется по формуле $R=\sum\_{i=1}^{M}\left|X\_{i}-Y\_{i}\right|$. При этом, чем меньше R, тем более похожими являются вектора.

Произвести разделение векторов на K групп, в каждой из которых должны находится вектора с максимальной степенью схожести (задача кластеризации).

Деление произвести по следующим правилам:

1.1. Рассчитывается матрица евклидовых расстояний между всеми парами объектов;

1.2. В цикле просматриваются все пары из N векторов и определяется пара, наиболее близкая по расстоянию. Объекты, входящие в эту пару, помечаются как выбранные.

1.3. Операция 1.2 повторяется до тех пор, пока все объекты не станут помеченными.

1.4. В результате получится массив, содержащий цепочку пар объектов и расстояний между ними (т. е. его элементами являются три числа: номера пары объектов и расстояние между ними).

1.5. В этом массиве найти элементы с (K-1)-первыми максимальными расстояниями.

1.6. В цикле: объекты с первого до первого максимального расстояния отнести в первую группу, объекты с первого максимального расстояния до второго максимального расстояния отнести во вторую группу и т.д.

1. Задан массив, состоящий из N векторов размерности M. Мера схожести векторов X и Y определяется по формуле $R=\sum\_{i=1}^{M}\left|X\_{i}-Y\_{i}\right|$. При этом, чем меньше R, тем более похожими являются вектора.

Произвести разделение векторов на K групп, в каждой из которых должны находится вектора с максимальной степенью схожести (задача кластеризации).

Деление производится по следующим правилам:

а) Все N объектов случайным образом размещаются в K группах;

б) В каждой группе определяются средние арифметические попавших туда объектов для каждой размерности. В результате для каждой группы формируется центроидный вектор размерностью M;

в) В цикле просматриваются все N векторов и для каждого из них ищется максимально близкий центроид и такой вектор перемещается в соответствующую группу;

г) операции а) и б) выполняются до тех пор, пока не прекратятся перемещения векторов между группами.

1. Задан массив, состоящий из N векторов размерности M. Мера схожести векторов X и Y определяется по формуле $R=\sum\_{i=1}^{M}\left|X\_{i}-Y\_{i}\right|$. При этом, чем меньше R, тем более похожими являются вектора.

Произвести разделение векторов на группы, в каждую из которых попадают вектора с максимальной степенью схожести (задача кластеризации).

Деление производится по следующим правилам:

а) берется первый вектор и помещается в первую группу;

б) среди оставшихся ищется пара наиболее близких векторов. Если один из наиболее близких векторов уже находится в какой-то группе, то второй вектор помещается в эту же группу. Если ни один из найденных векторов не находится ни в какой группе, то они оба помещаются в новую группу;

в) операция б) выполняется до тех пор, пока не буду размещены все вектора.

1. Задан массив, состоящий из N векторов размерности M. Произвести разделение векторов на группы, используя гравитационный алгоритм. Суть метода заключается в следующем:

- каждый вектор рассматривается как некая точка в M–мерном пространстве;

- эти точки взаимодействуют между собой по правилам (формуле) всемирного тяготения, т. е. каждая точка притягивает все остальные и притягивается ко всем остальным;

- алгоритм является итерационным (т. е. выполняется по шагам). На каждом шаге для каждого объекта рассчитывается направление (градиент) сдвига и объект сдвигается в этом направлении на некоторую величину.

- в результате близко расположенные объекты стягиваются в одну группу.

- вычисления заканчиваются при условии, что в какой-то из групп, расстояние между какими-то объектами становится меньше заданной величины. Величина минимального расстояния обычно задается исходя из возможностей системы программирования

1. **Совместные множества**

Задано N товаров (примерный перечень: Молоко, Сметана, Творог, Кефир, Сыр, Йогурт, Ряженка, Хлеб черный, Хлеб белый, Мясо, Колбаса, Рыба, Курица, Индейка, Конфеты, Печенье, Шоколад, Мармелад, Зефир, Овощи, Фрукты, Алкоголь, Табак) и массив покупок (размерности M). В одной покупке может быть несколько товаров из списка. Произвести разделение товаров на группы, в которые включены товары, наиболее часто покупаемые совместно.

1. Во многих магазинах накапливается информация о продажах товаров с течением времени. При этом в первую очередь регистрируются такие данные как дата продажи, наименование товара и идентификатор клиента. Эти данные можно использовать для выявления наиболее часто встречающихся последовательностей покупок. Например, если человек покупает стационарный компьютер, то затем он с высокой вероятностью купит принтер, сканер и постоянно начнет покупать расходные материалы для принтера.

Необходимо оценить какой товар с максимальной вероятностью купит каждый постоянный клиент при очередном заходе в магазин. Примерный перечнь товаров: (Компьютер, Принтер, Сканер, Ноутбук, Web-камера, Тонер, Флешка, Наушники, Колонки, Оперативная память, Внешний диск, Кабели аудио, Кабели USB, Кабели и шлейфы). Исходными данными для расчетов могут служить специально сгенерированные данные о продажах.

1. **Ассоциации**

Во многих интернет магазинах требуется регистрация пользователей. Помимо всего прочего, данные регистрации могут быть использованы в следующих целях. При входе на сайт магазина пользователю предоставляется возможность просмотреть каталоги товаров и цены на них. При этом количество товаров может быть настолько велико, что некоторым пользователям может просто не хватить терпения листать список в поисках нужного ему товара – и он просто уйдет с сайта, даже если нужный ему товар здесь есть и он самый дешевый и качественный. Чтобы этого не произошло осуществляется сужение области поиска, используя параметры пользователя.

Например.

- пользователь: девушка, 18 лет, студентка, не замужняя – с высокой вероятностью закажет косметику и/или кондитерские изделия;

- пользователь мужчина, 30 лет, оффисный работник, не женатый - с высокой вероятностью закажет алкогольные и/или табачные изделия;

и т.д.

Т.е. в сокращенный список товаров должны быть включены те товары, которые с наибольшей вероятностью купит пользователь, с характерными для него параметрами.

Составить программу, которая по введенным параметрам пользователя выводит список наиболее покупаемых им товаров.

1. **Гангстеры**

*N* гангстеров собираются в ресторан. *i*-й гангстер приходит в момент времени *Ti* и имеет богатство *Pi*. Дверь ресторана имеет *K* + 1 степень открытости, они обозначаются целыми числами из интервала [0, *K*]. Степень открытости двери может изменяться на единицу в единицу времени, то есть дверь может открыться на единицу, закрыться на единицу или остаться в том же состоянии. В начальный момент времени дверь закрыта (степень открытости 0). *i*-й гангстер заходит в ресторан, только если дверь открыта специально для него, то есть когда степень открытости двери соответствует его полноте *Si*. Если в момент, когда гангстер подходит к ресторану, степень открытости двери не соответствует его полноте, он уходит и больше не возвращается. Ресторан работает в интервале времени [0, *T*]. Требуется собрать гангстеров с максимальным суммарным богатством в ресторане, открывая и закрывая дверь соответствующим образом.


**10. Лабиринт.**

Лабиринт представляет собой прямоугольник, состоящий из N\*M сегментов. Каждый из сегментов может быть либо пустым, либо заполненным монолитной каменной стеной. Гарантируется, что левый верхний и правый нижний сегменты пусты. Лабиринт обнесён сверху, снизу, слева и справа стенами, оставляющими свободными только левый верхний и правый нижний углы. Директор лабиринта решил покрасить стены лабиринта, видимые изнутри (см. рисунок).

Помогите ему рассчитать количество краски, необходимой для этого.

**11. Конкурс 1**

В связи с внезапно возникшей вспышкой неизвестного заболевания министерство образования Тридевятого царства решило перевести студентов всех вузов на домашнюю форму обучения. Кроме того, очередной приём в вузы также запланировали провести дистанционно. Таким образом, абитуриенты теперь будут подавать документы на поступление через Тридевятую компьютерную сеть. Предполагается, что все смогут подать заявления на интересующие их специальности, не выходя из дома, а подведение итогов конкурса произойдёт автоматически, когда завершится подача заявлений.

Каждый вуз подводит итоги конкурса независимо друг от друга. Для этого в вузе составляются конкурсные списки абитуриентов по каждой специальности, ранжированные по убыванию суммы их баллов по трём предметам. В случае равенства суммы баллов производится дополнительное ранжирование по убыванию баллов за 1-й предмет, затем за 2-й, затем за 3-й. Если выявляется несколько абитуриентов с абсолютно одинаковыми баллами, проходящих на специальность, где свободных мест осталось меньше их количества - принимаются все такие абитуриенты.

Кроме того, в каждой записи указывается, какой приоритет для абитуриента имеет эта специальность. Согласно порядку приёма, абитуриент может подать заявления не более, чем на 3 специальности в один вуз. Требуется написать программу, которая по заданным конкурсным спискам определит списки зачисленных на каждую специальность. Входные данные: в первой строке - натуральное число N <= 50 - количество специальностей в вузе. Далее - N описаний конкурсных списков по каждой специальности. Описание конкурсного списка начинается с заголовка - строки, где указан шифр или название специальности - строка из цифр, латинских букв, знаков подчёркивания, точек и запятых, длина которой не превышает 50 символов, и через пробел - количество мест для приёма. Количество мест - натуральное число, не превышающее 200. Далее идут строки, в каждой из которых указана информация об очередном абитуриенте. Строка с описанием абитуриента состоит из его имени - слова, записанного латинскими буквами, длина которого не превышает 20; затем через пробел 3 оценки - натуральных числа, не превышающих 100; затем натуральное число от 1 до 3 – приоритет данной специальности согласно предпочтению абитуриента.

Описание каждого конкурсного списка заканчивается пустой строкой. Гарантируется, что у каждого абитуриента приоритеты не повторяются и номер приоритета не превышает количество выбранных им специальностей. Баллы каждого абитуриента одинаковы на всех специальностях. Выходные данные: списки поступивших (зачисленных) на каждую специальность. Порядок специальностей должен соответствовать исходному. В первой строке каждого списка указывается её название и через пробел - количество зачисленных; далее в каждой строке - имя абитуриента. Имена абитуриентов следует отсортировать по возрастанию в алфавитном порядке.

12. **Конкурс 2**

Требуется решить задачу 11 с тем отличием, что наборы экзаменов на разные программы различаются, а абитуриенты имеют более трёх оценок. Для подведения итогов конкурса в вузе составляются конкурсные списки абитуриентов по каждой специальности, ранжированные по убыванию суммы их баллов по трём предметам. В случае равенства суммы баллов производится дополнительное ранжирование по убыванию баллов за 1-й предмет, затем за 2-й, затем за 3-й. Если выявляется несколько абитуриентов с абсолютно одинаковыми баллами, проходящих на специальность, где свободных мест осталось меньше их количества - принимаются все такие абитуриенты. Кроме того, в каждой записи указывается, какой приоритет для абитуриента имеет эта специальность.

Согласно порядку приёма, абитуриент может подать заявления не более, чем на 3 специальности в один вуз.

Требуется написать программу, которая по заданным конкурсным спискам определит списки зачисленных на каждую специальность. Входные данные: в первой строке - натуральное число N <= 50 - количество специальностей в вузе. Далее - N описаний конкурсных списков по каждой специальности. Описание конкурсного списка начинается с заголовка - строки, где указан шифр или название специальности - строка из цифр, латинских букв, знаков подчёркивания, точек и запятых, длина которой не превышает 50 символов, количество мест для приёма и шифры трёх предметов, закодированные заглавными латинскими буквами. Элементы строки разделены пробелами.

Количество мест - натуральное число, не превышающее 200. Далее идут строки, в каждой из которых указана информация об очередном абитуриенте. Строка с описанием абитуриента состоит из его имени - слова, записанного латинскими буквами, длина которого не превышает 20; затем через пробел 3 оценки - натуральных числа, не превышающих 100; затем натуральное число от 1 до 3 - приоритет данной специальности согласно предпочтению абитуриента. Описание каждого конкурсного списка заканчивается пустой строкой.

Гарантируется, что у каждого абитуриента приоритеты не повторяются и номер приоритета не превышает количество выбранных им специальностей. Баллы каждого абитуриента по одному и тому же предмету одинаковы на всех специальностях.

Выходные данные: списки поступивших (зачисленных) на каждую специальность. Порядок специальностей должен соответствовать исходному.

В первой строке каждого списка указывается её название и через пробел

- количество зачисленных;

далее в каждой строке - имя абитуриента. Имена абитуриентов следует отсортировать по возрастанию в алфавитном порядке

**13. Сокровища**

Эльф, гном и человек шли по подземелью в поисках сокровищ. Уперевшись в стену, которая заграждала проход:

• гном пришёл к выводу, что это не стена, а дверь, которую заложили мастера древности, а за дверью их ждёт сокровище;

• человек нашёл на полу невообразимое количество фиолетовых, синих, жёлтых и красных камешков, но лишь один зелёный;

• эльф припомнил, что мастера древности любили загадки и математические равенства;
После вынужденного обсуждения также были выяснены следующие факты:

• на стену нанесены барельефы чисел;

• между числами находятся пазы (куда успешно помещаются найденные разноцветные камешки – по одному в паз);

• мастера древности рассматривали цвета как знаки арифметических операций:

o сложение – фиолетовым;

o умножение – красным;

o вычитание – синим;

o равенство – зелёным;

o целочисленное деление – жёлтым;

o остаток от деления – оранжевым;

• древняя культура использовала только целые числа.

На вход программе подаётся строка с целыми числами, нанесённых на стену, расставьте камни так, чтобы получилось математическое равенство.

Ответ запишите строкой, указывая через пробел символы (+, -, \*, / – целочисленное деление, % – остаток от деления; операции указаны в порядке убывания частоты использования) арифметических операций, которые символизируются нужными цветами. В случае, если таких записей несколько укажите вариант с наиболее часто употребляемыми операциями.

Входная строка представляет собой последовательность чисел, записанные через произвольное число пробелов.

Выходной строкой является строка, которую необходимо найти по заданию. Изменять формат входных и выходных данных запрещено.

**14. Анализ хромосом**

При анализе генома человека иногда есть необходимость находить повторяющиеся последовательности в строке, состоящей из алфавита A,C,G,T. Необходимо составить алгоритм, работающий для произвольного генома, определяющий в заданной строке наиболее длинную подстроку, которая повторяется заданное количество раз (или вернуть пустую строку).

Входная строка представляет собой число повторений и заданная строка, записанные через произвольное число пробелов. Выходной строкой является строка, которую необходимо найти по заданию

**15. Шарики**

Петя вспомнил о коробке с цветными шариками и стал придумывать игру. Все шарики выставил в линию. И стал рассматривать шарики слева направо. Когда обнаруживал непрерывную цепочку из трех и более шариков одного цвета, то удалял эти шарики из линии. Все оставшиеся шарики после удаления сдвигал друг к другу, и повторял описанную операцию до тех пор, пока было возможно. Игра занимала много времени, и он решил написать компьютерную программу.

Напишите и вы программу, которая определяет, сколько шариков будет удалено и какие шарики останутся.

**16. Волшебный Мост**

Крестьянин, возвращаясь с ярмарки, увидел на мосту странную картину. Какой-то человек сначала считал деньги в кошельке, затем бросал в реку несколько монеток, бежал на другой конец моста, снова считал деньги в кошельке, и опять бросал несколько монеток и шёл на другой конец моста. Наконец, пересчитав свои деньги, он явно обрадовался и отправился в дальнейший путь.

 Что ты делал? Зачем ты бросал деньги в воду? – спросил крестьянин, догнав странного человека.

Видя, что свой секрет скрыть не удастся, человек рассказал, что мост волшебный. Если бросить с моста ровно 29 копеек, то, как только пройдёшь мост, количество рублей в оставшейся сумме денег превращаются в новой сумме в количество копеек, а копейки – в рубли. Перейдя мост несколько раз, можно получить сумму, намного больше первоначальной.

- Самое важное – вовремя остановиться, - сказал человек и ушёл.

Крестьянин задумался, достал кошелёк и пересчитал свои деньги. У него было 46 рублей 47 копеек. «29 копеек – не деньги, дай-ка попробую». После первого прохода у него получилось 18 рублей 46 копеек, после второго прохода – 17 рублей 18 копеек, а после третьего – 89 рублей 16 копеек. «Ух-ты! А ещё больше можно получить?» - обрадовался крестьянин. После четвёртого прохода у него стало 87 рублей 88 копеек, после пятого – 59 рублей 87 копеек, после шестого – 58 рублей 59 копеек, после седьмого – 30 рублей 58 копеек, после восьмого – 29 рублей 30 копеек, после девятого – 1 рубль 29 копеек, а после десятого осталась 1 копейка.

«Эх, надо было после третьего раза остановиться!» - расстроился крестьянин.

Напишите программу, которая по начальной сумме денег у крестьянина и заданному количеству копеек, которое необходимо бросать с моста, определит оптимальное число проходов по мосту для получения наибольшей конечной суммы.

**17. Дороги**

Наступила осень, стали лить дожди. Дорога состоит из участков, последовательно пронумерованных целыми числами от 1 до N. К сожалению, иногда некоторые участки дорог затапливаются дождями, поэтому ремонт таких участков дорог невозможен. Изначально уровень воды на каждом участке дороги равен 0. Далее в течении k дней каждый день идет дождь.

Каждый дождь происходит только над случайным участком дороги, в результате чего уровень воды на нём увеличивается на 1. Если после этого уровень воды поднялся до 2, то вода начинает перетекать. На каждом участке дороги с номером i, где уровень воды был 2, он опускается до 0, а на участках с номерами i − 1 и i + 1 уровень воды увеличивается на 1. Все такие перетекания происходят одновременно.

Если после этого на каких-то участках уровень воды снова поднялся до 2, то процесс повторяется одновременно для всех участков.

Если участок является крайним (т. е. первым или последним), то утекающая за пределы дороги вода исчезает.

Определить состояние дороги на k-ый день осени.

18. Комната представляет собой прямоугольник размером N\*M. В ней на полу размещено k плиток, стороны которых параллельный сторонам комнаты. Координаты левого верхнего угла плитки равны xi, yi. Размер каждой плитки равен wi\*hi. Плитки могут перекрываться. Определить площадь не покрытой части пола.

1. Имеются файлы с текстами разных авторов. И имеется также текст неизвестного автора. Необходимо составить программу, определяющую какому из известных авторов принадлежит неизвестный текст с наибольшей вероятностью. Мера сходства обычно определяется исходя из сравнения частотных словарей разных авторов и частотного словаря неизвестного текста. Наиболее часто применяется так называемое косинусное сходство:

$$R=ArcCos(θ)=\frac{\sum\_{i=1}^{n}x\_{i}y\_{i}}{\sqrt{\sum\_{i=1}^{n}x\_{i}^{2}}\sqrt{\sum\_{i=1}^{n}y\_{i}^{2}}}$$

где n – количество слов в словаре;

 xi- частота встречаемости i-ого слова в первом тексте;

 yi- частота встречаемости i-ого слова во втором тексте.

Значение R интерпретируется как некоторый угол между векторами X и Y в многомерном пространстве. Если вектора совпадают, то угол между ними равен 0 и значение R будет равно 1. Если вектора не совпадают, то максимальный угол между ними будет равен 90 градусов и значение R будет равно 0.

1. Имеются файлы с текстами разных авторов. И имеется также текст неизвестного автора. Необходимо составить программу, определяющую какому из известных авторов принадлежит неизвестный текст с наибольшей вероятностью. Мера сходства обычно определяется исходя из сравнения частотных словарей разных авторов и частотного словаря неизвестного текста. Наиболее часто применяется так называемое косинусное сходство:

$$R=ArcCos(θ)=\frac{\sum\_{i=1}^{n}x\_{i}y\_{i}}{\sqrt{\sum\_{i=1}^{n}x\_{i}^{2}}\sqrt{\sum\_{i=1}^{n}y\_{i}^{2}}}$$

где n – количество слов в словаре;

 xi- частота встречаемости i-ого слова в первом тексте;

 yi- частота встречаемости i-ого слова во втором тексте.

Значение R интерпретируется как некоторый угол между векторами X и Y в многомерном пространстве. Если вектора совпадают, то угол между ними равен 0 и значение R будет равно 1. Если вектора не совпадают, то максимальный угол между ними будет равен 90 градусов и значение R будет равно 0.