

MODULE 1

Уровень 1

1. Дан массив чисел. Вернуть число уникальных символов

Пример: для 1 4 5 1 1 3 ответ 4 .

2. Ход коня по бесконечной шахматной доске. Вводится текущее положение коня и клетка в которую пробуют его передвинуть за 1 ход. Программа должна проверить, возможно ли это и сделать.

3. Даны точки A, B, C на плоскости. Вычислить площадь треугольника ABC.

Уровень 2

1. Учитывая строку, содержащую символы '(', ')', '{', '}', '[', ']', определите, является ли входная строка допустимой.

Входная строка действительна, если:

Открытые скобки должны быть закрыты скобками того же типа.

Открытые скобки должны быть закрыты в правильном порядке.

Обратите внимание, что пустая строка также считается допустимой.

2. По заданному бинарному дереву найдите его максимальную глубину.

Максимальная глубина - это количество узлов вдоль самого длинного пути от корневого узла до самого дальнего конечного узла.

/**

** Definition for a binary tree node.*

** public class TreeNode {*

** int val;*

** TreeNode left;*

** TreeNode right;*

** TreeNode(int x) { val = x; }*

** }*

Уровень 3

Согласно статье в Википедии: «Игра Жизни, также известная просто как Жизнь, представляет собой клеточный автомат, созданный британским математиком Джоном Хортоном Конвеем в 1970 году». При наличии доски с размерностью m на n ячеек каждая ячейка имеет начальное состояние live (1) или dead (0).

Каждая ячейка взаимодействует со своими восемью соседями (по горизонтали, вертикали и диагонали), используя следующие четыре правила (взяты из вышеупомянутой статьи в Википедии): Любая живая ячейка с менее чем двумя живыми соседями умирает, как если бы она была вызвана недостаточным населением. Любая живая клетка с двумя или тремя живыми соседями доживает до следующего поколения. Любая живая клетка с более чем тремя живыми соседями умирает, как если бы она была перенаселена. Любая мертвая клетка с ровно тремя живыми соседями становится живой клеткой, как если бы она воспроизводилась. Напишите функцию для вычисления следующего состояния (после одного обновления) платы с учетом ее текущего состояния. Следующее состояние создается путем применения вышеуказанных правил одновременно к каждой ячейке в текущем состоянии, где рождения и смерти происходят одновременно.

MODULE 2

1. Дан список дат (строковая запись) в форматах типа "2020/04/05", "05/04/2020", "04-05-2020" (все даты в примере - 5е апреля 2020)
Вернуть список дат (строковая запись) в формате "20200405". Даты с неверным форматом - игнорировать.
2. Дан список имен. Найти первое уникальное имя. Допустимая временная сложность - $O(n)$ при условии, что доступ к элементу списка по индексу - $O(1)$.
3. Дан список городов. Каждый путь между городами имеет цену (целое положительное число). Задача - найти самый выгодный путь между двумя городами. Максимально возможная цена пути - 200000.

Данные условия необходимо считать из файла input.txt
Результат надо записать в файл output.txt

Ввод

n [количество городов ≤ 10000]
NAME [имя города]
p [количество соседей города *NAME*]
nr cost [*nr* - индекс соседа *NAME* (начиная с 1)]
 [*cost* - стоимость пути]
r [количество путей, которые надо найти ≤ 100]
NAME1 NAME2 [*NAME1* - начало пути, *NAME2* - конец пути]

Вывод

cost [минимальная стоимость пути от *NAME1* до *NAME2*]

Пример

Ввод:

```
4
gdansk
2
2 1
3 3
bydgoszcz
3
1 1
3 1
4 4
torun
3
1 3
2 1
4 1
warszawa
2
2 4
3 1
2
gdansk warszawa
bydgoszcz warszawa
```

Вывод:

```
3
2
```

MODULE 3

У каждого пользователя может быть несколько счетов, и он может проводить операции по одному из них.

Операция может иметь в себя категорию доходов, если операция приносит прибыль, либо категорию расходов в противном случае.

Операции с нулевым оборотом недопустимы.

Операции без категории или с не валидной для типа операции (доход/расход) категорией недопустимы.

Приложение должно хранить информацию о всех вышеописанных сущностях в БД, а также хранить лог действий пользователя в локальном файле.

Необходимо реализовать два режима работы приложения:

- 1) Добавление новой операции существующим пользователем
- 2) Экспорт выписки по счету в csv формате (должен создавать выходной файл .csv).

Выписка должна включать список операций за определенный период.

Первый режим реализовать при помощи JPA/Hibernate, второй - при помощи JDBC

В обоих случаях, пользователь должен передать в приложение, посредством аргументов командной строки или переменных среды: идентификатор пользователя (его id, email, или phone number, на выбор разработчика), а также имя пользователя и пароль для подключения к БД.

Для дат в БД использовать тип timestamp, в java - Instant. Текстовый формат дат - ISO_LOCAL_DATE_TIME из java.time.

Данные для тестирования ввести в БД можно вручную (пользователи, счета и категории операций)