**JS + html + CSS**

*Инструкция к заданиям 1 - 26:*

Ввод данных реализуете через html-страничку. Например, если на вход подается число или строка, можете использовать инпут для ввода данных. Естественно, при нажатии на кнопку, программа выполняется и результат отображается на страничке. Если на вход подается матрица, можно реализовать таблицу в каждой ячейки которой будет инпут, границы которого будут не отображаться. Аналогично можно поступать и с выводом матрицы на страницу. Если на вход подается последовательность, можно ввести числа в инпут через пробел, как вариант. Выход - список или текстовое поле.

*Инструкция к заданиям 27 - 35:*

Ввод данных реализуете через html-страничку. Например, если есть класс Student с различными атрибутами, можно эти атрибуты вводить через инпуты (если это требуется в задаче). Также, будут кнопки соответствующие методам, которые вам нужно реализовать (getSalary() - рассчитать стипендию, при нажатии на кнопку, тут же появляется результат).

*Дополнительная инструкция указана около задания.*

*Распределение по вариантам приложено в следующем файле.*

**Задания:**

1. Вводятся целые k, m. Вывести целые числа между k и m, которые делятся на 3.
2. Вычислить квадратный корень числа x с точностью до 6 знака после запятой. Не использовать Math.
3. Вводится n чисел. Проверить, что среди них существует ровно два таких числа, что длина (количество цифр) каждого из них равна 3 или 5, а их цифры либо все четные, либо все нечетные.
4. Вводится n чисел. Проверить, что существует ровно три числа, в котором цифры идут по убыванию.
5. Локальным максимумом в последовательности назовем элемент, который больше предыдущего и следующего (если они есть). Проверить, что во входной последовательности существует четный по значению локальный максимум.
6. Проверить, что во входной последовательности целых чисел существует ровно два четных по значению локальных максимума.
7. Вводится целое 2 < k < 9, затем вводится целое число n, которое можно интерпретировать как число в k-ичной системе счисления. Сконвертировать n в десятичную систему счисления. Строковые типы не используются.
8. Вводится строка, представляющая собой слова на английском языке, записанные через пробел. Подсчитать, сколько слов начинаются на заглавную букву.
9. Парни и девушки идут в кинотеатр. На сеансе они договариваются сидеть так, чтобы: девушки не сидели по краям, три парня не должны сидеть рядом. Вводится n и k - общее число друзей и количество девушек. Вывести хотя бы одну возможную рассадку или написать, что такой рассадки нет.
10. Служба доставки владеет грузовиками, которые могут развозить 150 кг посылок. Однако сами посылки имеют вес либо 100 кг, либо 50 кг. На вход подается массив целых чисел, каждое число - либо 50, либо 100. Необходимо вывести минимальное количество поездок, которое надо совершить, чтобы отвезти все введенные посылки.
11. Реализовать модель игры – игроки последовательно друг другу наносят удары силой от 1 до 9, при этом у того, кого ударили, из очков здоровья (health points, hp) отнимается сила удара. Игра заканчивается, когда hp одного из игроков стало <= 0. В задаче должны быть классы Игрок и Игра, в которой весь процесс происходит. У каждого игрока есть имя и hp. Процесс игры должен принадлежать объекту класса Игра.

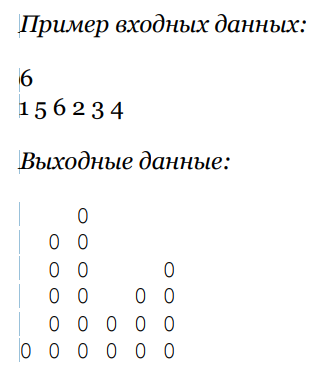
Как это должно выглядеть на html-страничке: страница разделена на две части. Слева и справа заголовки, содержащие надпись Игрок 1 и Игрок 2. У каждого из игроков есть инпут, в который они записывают силу удара (не забудьте сделать проверку на силу удара в -3 или в 100). Также, у каждого игрока снизу подписано hp, которое будет обновляться после нанесения атаки или принятия атаки.

1. Имеется калькулятор, который выполняет три операции:

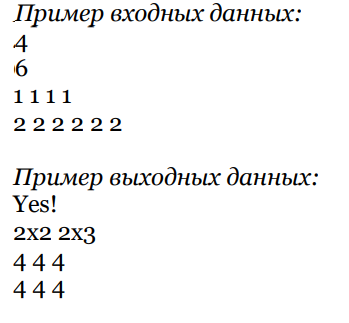
* Прибавить к числу X единицу;
* Умножить число X на 2;
* Умножить число X на 3.

Определите, какое наименьшее число операций необходимо для того, чтобы получить из числа 1 заданное число N. Выведите это число, и, на следующей строке, набор исполненных операций вида «111231».

1. Вводятся два массива, представляющие собой подстановку. Подстановка например:  
   (0,1,2,3,4)  
   (4,1,3,0,2)  
   где первая строка - индексы, вторая - элементы массива. Умножить одну подстановку на другую, результат записать в первый массив, результат вывести списком на html-страничку.
2. Подсчитать в двумерном массиве максимум по суммам элементов на главных диагоналях.  
   Пример  
   1 2 3  
   1 2 3  
   0 7 2  
   Главные диагонали: 1 2 2, 2 3 0, 3 1 7.   
   Суммы: 5, 5, 11. Максимум 11
3. Проверить, что в двумерном массиве каждая сумма элементов на побочных диагоналях является четной.
4. Вводится целочисленная квадратная матрица размера 2n+1. Заменить нулями элементы, расположенные на верхнем и нижнем треугольниках, образованных пересечением главной и побочной диагоналей. Правый и левый треугольники не трогать. Пример:  
     
   Вход:  
   1 2 3 4 5  
   2 3 4 1 2  
   0 3 2 3 1  
   9 2 3 1 4   
   3 8 0 8 6   
   Выход:  
   1 0 0 0 5  
   2 3 0 1 2  
   0 3 2 3 1  
   9 2 0 1 4   
   3 0 0 0 6
5. Вводится квадратная матрица размера n. Привести ее к треугольному виду и вывести на страничку. Не забудьте, что надо обрабатывать случай поиска ненулевого элемента в столбце (поиск – существование – квантор).
6. Подсчитать в двумерном массиве максимум по суммам элементов на главных диагоналях.  
   Пример  
   1 2 3  
   1 2 3  
   0 7 2  
   Главные диагонали: 1 2 2, 2 3 0, 3 1 7.   
   Суммы: 5, 5, 11. Максимум 11
7. Реализовать с помощью двумерных массивов умножение матриц.
8. Вводится две строки. Написать функцию, сравнивающую строки в лексикографическом порядке (кто раньше, кто позже по алфавиту). Из строковых методов использовать только length и charAt.
9. Вводится массив строк. С помощью написанной функции выше отсортировать строки в лексикографическом порядке.
10. Вводится строка, представляющая собой английский текст. Подсчитать и вывести, сколько раз встречается каждая буква английского алфавита в тексте. Не различать регистр (‘A’ и ‘a’ – одна и та же буква).
11. Вводится строка, представляющая собой английский текст. Каждое второе вхождение строки “true” заменить на “false”. Не использовать методы строки, кроме charAt и length().
12. Вводится строка, представляющая собой английский текст. Каждое вхождение слова «mom” заменить на “dad”. Не использовать методы строки, кроме charAt и length().
13. Вводится число n >= 1, затем массив из n целых положительных чисел. Построить по нему гистограмму: разделенные пробелами, выровненные по нижней границе столбцы, состоящие из 1 символа, высота которых равна соответствующим элементам массива. Дополнительные массивы не использовать.



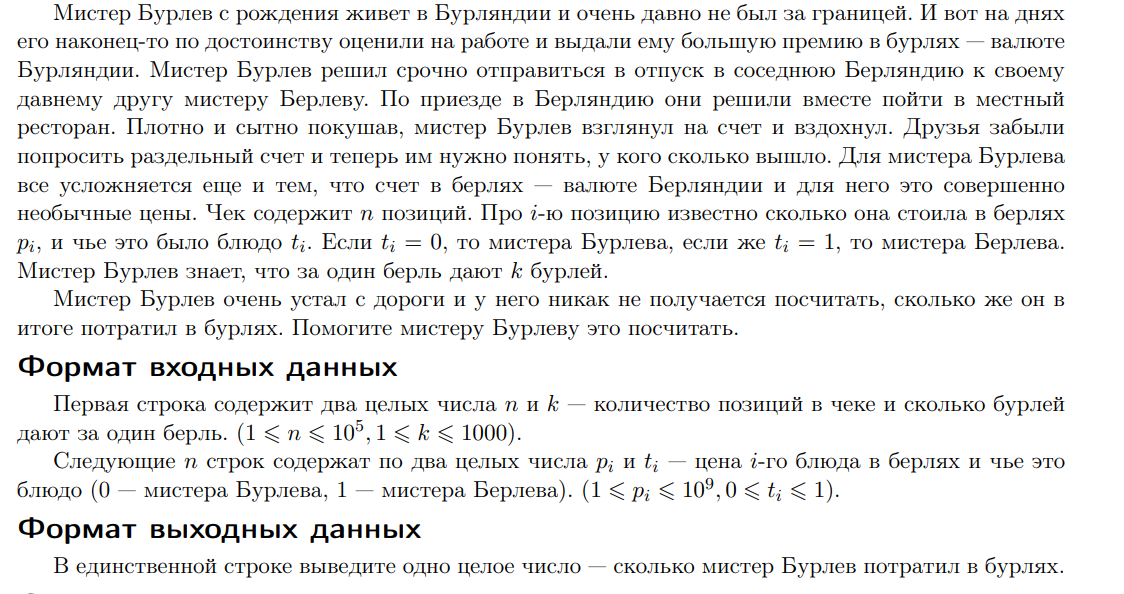
1. Любую матрицу можно представить как одномерный массив, просто переписав построчно все элементы матрицы в одну последовательность. Итак, вводятся числа m и n. Затем целочисленные массивы размера m и n. Выяснить, могут ли эти массивы представлять матрицы, которые можно перемножить. По-другому: существуют ли такие целые a, b, c, что a \* b = m и b \* c = n? Если ответ на вопрос выше положительный, то вывести размеры обеих матриц, выполнить умножение матриц. Результат положить в одномерный массив размера m x n, вывести его как таблицу. Дополнительных массивов не использовать.

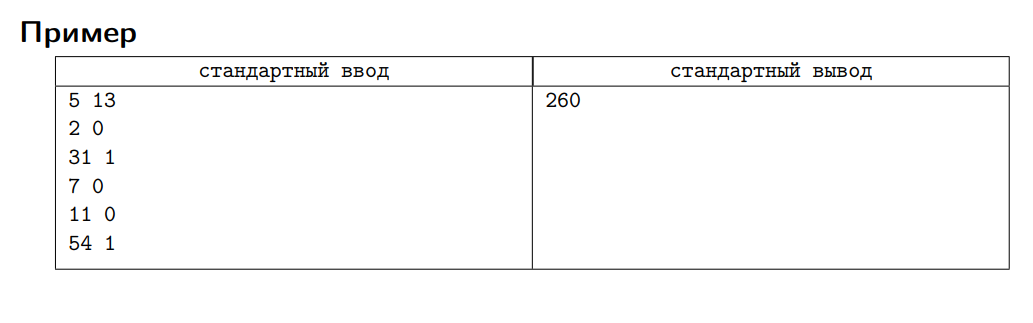


1. Реализуйте класс Worker (Работник), который будет иметь следующие свойства: name (имя), surname (фамилия), rate (ставка за день работы), days (количество отработанных дней). Также класс должен иметь метод getSalary(), который будет выводить зарплату работника. Зарплата - это произведение (умножение) ставки rate на количество отработанных дней days.
2. Реализуйте класс MyString, который будет иметь следующие методы: метод reverse(), который параметром принимает строку, а возвращает ее в перевернутом виде, метод ucFirst(), который параметром принимает строку, а возвращает эту же строку, сделав ее первую букву заглавной и метод ucWords, который принимает строку и делает заглавной первую букву каждого слова этой строки.
3. Реализуйте класс Validator, который будет проверять строки. К примеру, у него будет метод isEmail параметром принимает строку и проверяет, является ли она корректным емейлом или нет. Если является - возвращает true, если не является - то false. Кроме того, класс будет иметь следующие методы: метод isDomain для проверки домена, метод isDate для проверки даты и метод isPhone для проверки телефона.
4. Реализуйте класс Student (Студент).. Этот класс должен иметь следующие свойства: name, surname, year (год поступления в вуз). Класс должен иметь метод getFullName() , с помощью которого можно вывести одновременно имя и фамилию студента. Также класс должен иметь метод getCourse(), который будет выводить текущий курс студента (от 1 до 5). Курс вычисляется так: нужно от текущего года отнять год поступления в вуз. Текущий год получите самостоятельно.
5. Создать класс Concert - музыкальный концерт, который имеет атрибуты - city (город проведения), date (дата проведения), author (исполнитель), genre (жанр). Реализуйте следующие методы:

* Concert(city, date, author, genre) - конструктор, возвращающий объект Concert;
* concerts[] SortAuthor(concerts[]) - сортирует список концертов по алфавитному порядку исполнителей;
* concerts[] SortGenre(concerts[]) - сортирует список концертов по алфавитному порядку жанров;
* concerts[] SortCities(concerts[]) - сортирует список концертов по алфавитному порядку городов;
* concerts[] deleteLastCon(concerts[]) - удаляет из списка те концерты, которые уже прошли;
* void getList(concerts[]) - выводит на экран список всех концертов;
* void getListNewDate(concerts[]) - выводит на экран список только тех концертов, которые еще не прошли;
* void getListLastDate(concerts[]) - выводит на экран список только тех концертов, которые уже прошли;
* concerts[] getListByDate(String date) - возвращает список концертов, которые будут проходить в заданную дату (date).

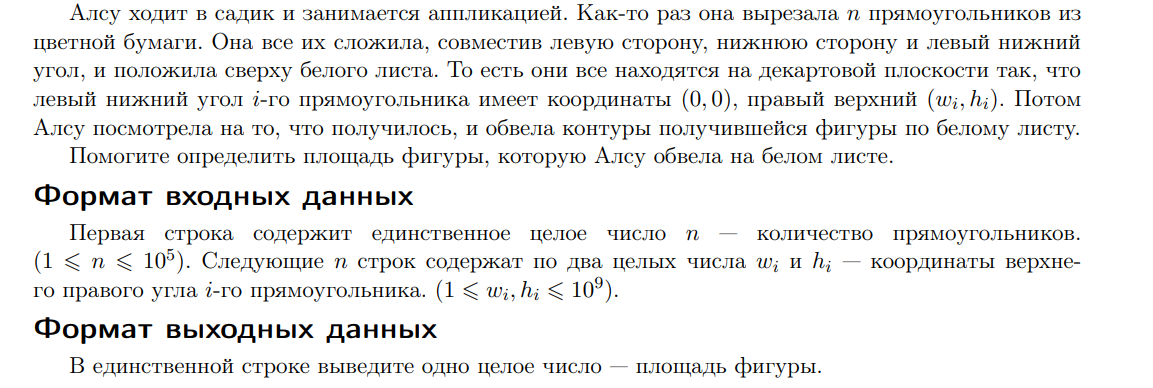
1. Класс RationalFraction - рациональная дробь. Атрибуты - два целых числа (числитель и знаменатель). Методы:  
   • RationalFraction() - конструктор для дроби, равной нулю;  
   • RationalFraction(int, int) - конструктор дроби со значениями числителя и знаменателя; в конструкторах устраняйте дублирование кода;  
   • void reduce() - сокращение дроби;  
   • RationalFraction add(RationalFraction) - сложение дроби с другой дробью, результат возвращается как новый объект (не забудьте сократить)  
   • RationalFraction sub(RationalFraction) - вычитание из дроби другой дроби, результат возвращается как новый объект (не забудьте сократить);  
   • RationalFraction mult(RationalFraction) - умножение дроби на другую дробь, результат возвращается как новый объект (сократить)  
   • RationalFraction div(RationalFraction) - деление дроби на другую дробь, результат возвращается как новый объект (сократить)  
   • String toString() - строковое представление дроби (например, -2/3);  
   • double value() - десятичное значение дроби;  
   • boolean equals(RationalFraction) - сравнить дробь с другой дробью   
   • (не забывайте, что 2/4 и 1/2 - одна и та же дробь)  
   • int numberPart() - целая часть дроби;
2. Класс Vector2D - двумерный вектор. Атрибуты - два вещественных числа (координаты).   
   • Vector2D() - конструктор для нулевого вектора;  
   • Vector2D(double, double) - конструктор вектора с координатами; в конструкторах устраняйте дублирование кода;  
   • Vector2D add(Vector2D) - сложение вектора с другим вектором, результат возвращается как новый объект.  
   • Vector2D sub(Vector2D) - вычитание из вектора другого вектора, результат возвращается как новый объект;  
   • Vector2D mult(double) - умножение вектора на вещественное число, результат возвращается как новый объект;  
     
   • String toString() - строковое представление вектора;  
   • double length() - длина вектора;  
   • double scalarProduct(Vector2D) - скалярное произведение вектора на другой вектор;  
   • double cos(Vector2D) - косинус угла между этим и другим вектором;  
   • boolean equals(Vector2D) - сравнить вектор с другим вектором;
3. Создать класс ComplexNumber - комплексное число. Атрибуты - действительная и мнимая части (два числа). Методы:  
   • ComplexNumber() - конструктор для нулевого комплексного числа;  
   • ComplexNumber(double, double) - конструктор комплексного числа с заданными значениями вещественной и мнимой части; в конструкторах устраняйте дублирование кода;  
   • ComplexNumber add(ComplexNumber) - сложение комплексного числа с другим комплексным числом;  
   • CompexNumber sub(ComplexNumber) - вычитание из комплексного числа другого комплексного числа;  
   • ComplexNumber multNumber(double) - умножение комплексного числа на вещественное число;  
   • ComplexNumber mult(ComplexNumber) - умножение комплексного числа на другое комплексное число;  
   • ComplexNumber div(ComplexNumber) - деление на другое комплексное число;  
   • double length() - модуль комплексного числа;   
   • String toString() - строковое представление комплексного числа. Только без всяких "2 \* i + - 3". Проверяйте знаки, чтобы было красиво: 2 \* i - 3.  
   • boolean equals(ComplexNumber) - сравнить комплексное число с другим комплексным числом;
4. Создать класс Matrix2x2 - двумерная матрица из вещественных чисел. Аргументы - содержимое матрицы (лучше, разумеется, хранить двумерным массивом, а то замучаетесь). Методы:  
   • Matrix2x2() - конструктор для нулевой матрицы;  
   • Matrix2x2(double) - конструктор для матрицы, у которой каждый элемент равен поданному числу;  
   • Matrix2x2(double [][]) - конструктор для матрицы, содержимое подается на вход в виде массива;  
   • Matrix2x2 add(Matrix2x2) - сложение матрицы с другой;  
   • Matrix2x2 sub(Matrix2x2) - вычитание из матрицы другой матрицы;  
   • Matrix2x2 multNumber(double) - умножение матрицы на вещественное число;  
   • Matrix2x2 mult(Matrix2x2) - умножение матрицы на другую матрицу;  
   • double det() - определитель матрицы;  
   • void transpon() - транспонировать матрицу;  
   • Matrix2x2 inverseMatrix() - возвратить обратную матрицу для заданной. Если это невозможно, вывести сообщение об ошибке и вернуть нулевую матрицу (кто вдруг знает исключения, может их использовать).  
   • Vector2D multVector(Vector2D) - умножить матрицу на двумерный вектор (считая его столбцом) и возвратить получившийся столбец в виде вектора.

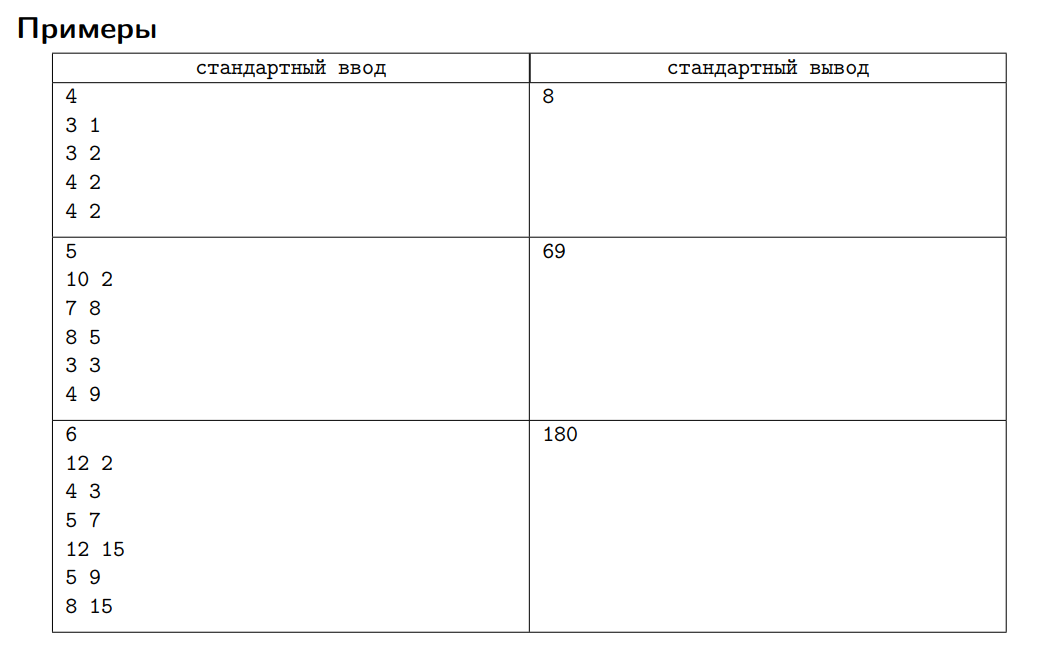
36.



*Здесь необходимо создать либо n-количество инпутов, либо считывать все данные через 1 инпут в виде строки.*

37.





*Здесь необходимо создать либо n-количество инпутов, либо считывать все данные через 1 инпут в виде строки.*

38. 