

**Заключительный (очный) этап научно-образовательного соревнования
Олимпиады школьников «Шаг в будущее» по профилю «Инженерное дело» специализации
«Техника и технологии» (общеобразовательный предмет информатика), весна 2020 г.**

11 класс

Вариант 1

Задача 1

Учительница математики Марина Григорьевна попросила старшеклассников составить программу для тестирования младших учеников по геометрии, которая будет выдавать задачи на проверку принадлежности точки - прямой.

Требуется написать программу, которая будет контролировать правильность ответов учеников.

Входные данные: в первой строке уравнение вида $ax+by+c=0$, где a , b и c - целые числа от 0 до 1000, а знаками операции могут быть как "+", так и "-". Во второй строке - целые числа - координаты проверяемой точки.

Выходные данные: если точка лежит на прямой - слово "YES", иначе - слово "NO" и через пробел - целая часть расстояния от точки до прямой.

Пример

Исходные данные	Результат
$1x+1y+0=0$ 1 -1	YES
$2x-3y+1=0$ 2 0	NO 1

Задача 2

В строке записано истинное логическое выражение с тремя переменными a , b и c . Над переменными применяются 2 операции: эквивалентность и исключающее «или». Требуется восстановить значения таблицы истинности функции, соответствующей этому выражению.

Исключающее «или» - булева функция двух переменных, результат которой истинен тогда и только тогда, когда один аргумент истинен, а второй - ложен.

Эквивалентность - логическое выражение, которое является истинным тогда, когда оба простых логических выражения (левая и правая части) имеют одинаковую истинность.

В рамках данной задачи будем обозначать исключающее «или» знаком " \wedge ", а эквивалентность - знаком " $=$ " без кавычек.

Входные данные: логическое выражение, состоящее из имён переменных a , b , c , знаков исключающего «или» и эквивалентности. Длина выражения не превышает 20 символов.

Выходные данные: 8 цифр 0 и 1, записанных неразрывно и означающих значения функции для каждой из комбинаций значений переменных:

0, 0, 0;

0, 0, 1;

0, 1, 0;

0, 1, 1;

1, 0, 0;

1, 0, 1;

1, 1, 0;

1, 1, 1.

Пример

Исходные данные	Результат
$a \wedge b \wedge c$	01101001
$a \wedge b = b \wedge c$	10100101

Примечание: операция исключающего «или» имеет более высокий приоритет по сравнению с эквивалентностью.

Задача 3

Дана запись двух больших целых чисел в шестнадцатеричной системе счисления и их произведения, при этом в записи допущена одна ошибка.

Написать программу, которая выявит допущенную ошибку и исправит её.

Входные данные: три строки, в каждой из которых записано по шестнадцатеричному числу. Количество знаков в каждом из чисел не превышает 100. Цифры от A до F записаны заглавными латинскими буквами.

Выходные данные: исправленная запись в том же формате.

Пример:

Исходные данные	Результат
10 A A2	10 A A0
1F 27 49A	1F 26 49A

Задача 4

Пете нравится решать sudoku, и он задумался, как можно изменить эту головоломку, чтобы она стала интереснее. После размышлений ему пришла идея сделать sudoku в 16-ричной системе счисления, но он засомневался, получится ли её решать так же, как обычную.

Правила 16-ричных sudoku, которые придумал Петя: дано поле 16x16, разделённое на квадраты 4x4. Допустимы все возможные 16-ричные цифры от 0 до F, при этом не должно быть повторов одной цифры в строках и столбцах поля, а также в отдельных квадратах.

Требуется написать программу, которая заполнит недостающие клетки в заданном 16-ричном sudoku.

Входные данные: 16 строк по 16 символов с цифрами от 0 до F. Пустые поля обозначены символами "_" (знак подчёркивания). Количество пустых полей не превышает 30.

Выходные данные: решённое sudoku - 16 строк по 16 символов с цифрами от 0 до F, такие, что все пустые поля заполнены пропущенными цифрами.

Пример:

Исходные данные	Результат
12_456_890ABCDEF 567890ABCDEF1234 90ABCDEF12345678 CDEF123_567890AB 3_567890A_CDE_12 7890ABCDEF123456 ABCDEF1234567890 EF12_4567890ABCD 678_0ABCDEF12345 234__7890A__DEF1 DEF1234567890_BC 0ABC_EF123456789 890ABCDEF123456_ _567890ABCDEF1_3 _1234567890ABCDE BCDEF123456789_A	1234567890ABCDEF 567890ABCDEF1234 90ABCDEF12345678 CDEF1234567890AB 34567890ABCDEF12 7890ABCDEF123456 ABCDEF1234567890 EF1234567890ABCD 67890ABCDEF12345 234567890ABCDEF1 DEF1234567890ABC 0ABCDEF123456789 890ABCDEF1234567 4567890ABCDEF123 F1234567890ABCDE BCDEF1234567890A

Примечание: гарантируется, что исходные данные корректны, sudoku может быть решено и решение единственное.

Задача 5

Двоичным деревом называется иерархическая структура данных, в которой каждый узел имеет не более двух потомков.

Двоичное дерево поиска - разновидность двоичного дерева, у которого оба поддерева (левое и правое) являются двоичными деревьями поиска; все узлы левого поддерева произвольного узла A меньше, чем значение самого узла A; все узлы правого поддерева произвольного узла A больше либо равны значению узла A.

Высотой узла называется максимальная длина нисходящего пути от этого узла к самому нижнему узлу, называемому листом. Высота корневого узла равна высоте всего дерева.

Дерево называется сбалансированным, если для любой его вершины высота левого и правого поддерева для этой вершины различаются не более чем на 1.

При прямом обходе дерева сначала обрабатывается ключ (значение) корня, затем - ключи левого и правого поддеревьев.

По заданным ключам вершин, полученным при прямом обходе, требуется определить, является ли дерево сбалансированным, а также определит высоту дерева.

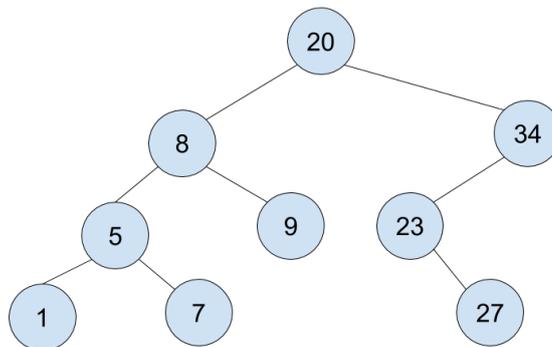
Входные данные: ключи всех вершин двоичного дерева поиска в порядке прямого обхода. Каждый ключ - натуральное число, не превышающее 105. Каждое значение задано в отдельной строке. Последняя строка состоит из одного символа - точки.

Выходные данные: первая строка - слово YES, если дерево сбалансированное, и NO, если нет.

Вторая строка - число, соответствующее высоте дерева.

Пример

Исходные данные и результат соответствуют дереву, изображённому на рисунке:



Исходные данные	Результат
20	NO 4
8	
5	
1	
7	
9	
34	
23	
27	
.	