

Заключительный этап, 9 класс (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (1 балл)

[XYZ_2024]

Дано выражение:

$$XYZ_9 + ZXY_{12} + YZX_{14} = 2024_{10}$$

В данном выражении X, Y и Z – допустимые различные цифры указанных систем счисления. Определите значения переменных Y и Z, если X = 5. В ответе укажите через пробел 2 целых положительных числа: сначала значение Y и затем значение Z.

2. Кодирование информации. Объем информации (2 балла)

[Canis aureus]

Для кодирования цветов часто используется RGB-палитра. В этом случае цвет каждого пикселя изображения кодируется с помощью трех отдельных числовых значений, называемыми цветовыми каналами (красный (R), зеленый (G) и синий (B)). Числовое значение по каждому цветовому каналу в общем случае может быть в диапазоне от 0 до 255. При сохранении каждого числового значения в памяти для него отводится минимальное, одинаковое для всех значений количество бит. Илья решил, что может хранить изображения используя меньше памяти.

Во-первых, он применил к каждому пикселю изображения следующий алгоритм сжатия:

1. Если значения всех цветовых каналов имеют одинаковую чётность - цвет не меняется.
2. Если только одно из значений нечётное - оно уменьшается на 1.
3. Если только одно из значений чётное - оно увеличивается на 1.

Во-вторых, он решил, что после сжатия будет отводить минимальное, одинаковое для всех значений количество бит не значению каждого цветового канала пикселя, а коду, который присваивается уникальной комбинации значений всех цветовых каналов каждого пикселя. Уникальной является комбинация, отличающаяся от любой другой значением хотя бы одного цветового канала.

Теперь Илья хочет оценить эффективность своей идеи. Определите, на сколько байт уменьшилось количество памяти, необходимое для хранения изображения размером 1920 на 1080 пикселей. Если необходимое количество памяти увеличилось – запишите в ответ отрицательное число. Например, если раньше требовалось 100 байт, а после преобразований потребуется 98 байт, то ответ будет равен 2, а если потребуется 102 байта, то ответ будет равен -2.

3. Основы логики (1 балл)

[Четыре импликации]

Два набора значений переменных A, B и C называют не эквивалентными, если значение хотя бы одной переменной различается.

Сколько существует не эквивалентных друг другу наборов значений переменных, при которых равенство выполняется при любом значении D:

$$(((D \rightarrow (A \wedge B \wedge C)) \rightarrow (A \wedge B \vee C)) \rightarrow (A \vee B \wedge C)) \rightarrow (A \vee B \vee C) = 1$$

В ответ укажите одно целое число.

4. Алгоритмизация и программирование. Формальные исполнители (2 балла)

[Бедная лошадка]

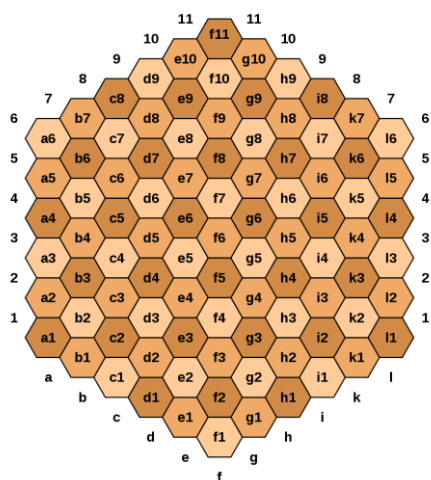
Недавно Ксюша узнала о существовании гексагональных шахмат. Больше всего её заинтересовали перемещения коней и слонов. Она задала набор дополнительных правил для перемещения коня:

1. Он может сделать сколько угодно корректных ходов.
2. Он не может наступать на клетки, находящиеся под боем у слонов (клетка находится под боем, если слон может попасть в неё за один корректный ход).
3. Он не может съесть слонов.
4. Слоны не могут двигаться.

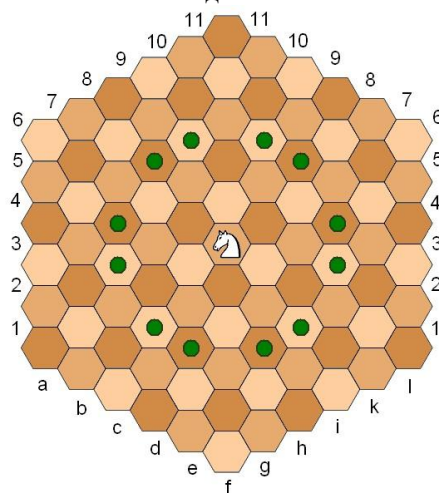
Ксюша решила определить, сколько есть клеток на поле, посещение которых конем не будет противоречить указанным условиям, но которые конь тем не менее не сможет посетить, если он будет начинать на клетке **e6**, а слоны будут стоять на клетках **d6**, **e9** и **i6**? В ответе напишите сначала количество клеток, а затем в возрастающем лексикографическом порядке через пробел эти клетки. Например, 3 a2 a3 b2. Если окажется, что таких клеток нет, напишите в ответ NULL.

Примечание:

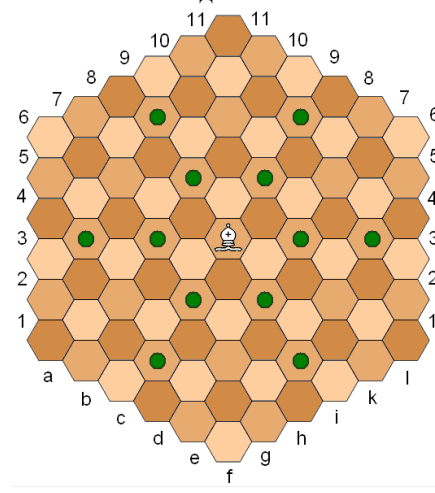
Нумерация клеток шахматной доски



Ход коня



Ход слона



Роль горизонталей выполняют косые линии полей, параллельные одной из неперпендикулярных сторон доски, а роль диагоналей — линии полей одного цвета

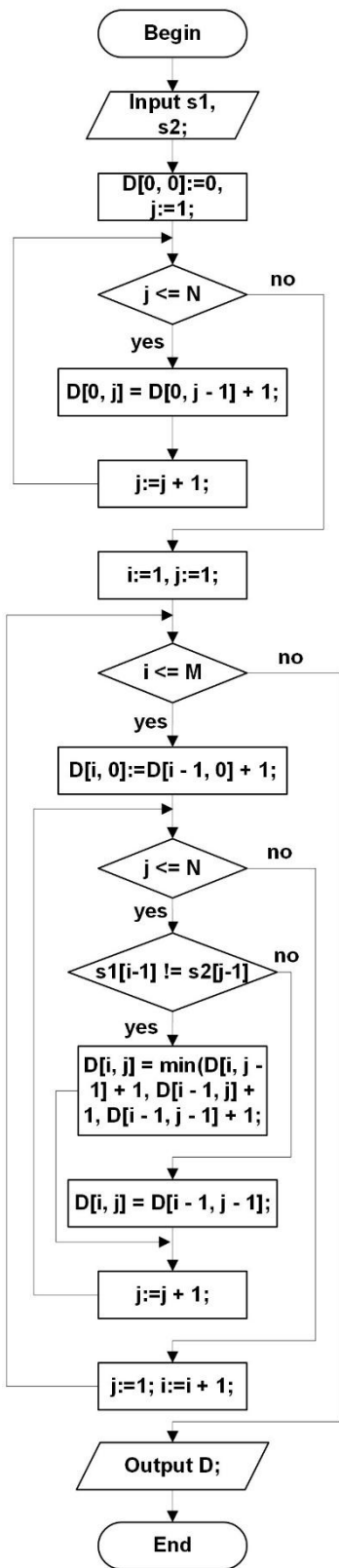
Конь ходит на два поля по вертикали или «горизонтали» и ещё на одно поле по другой вертикали или «горизонтали» (поворот на 120 градусов). Не обязательно, чтобы промежуточные клетки были свободны/достижимы, главное, чтобы были свободны начальная и конечная клетки.

Слон ходит на любое количество полей по диагоналям своего цвета.

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (3 балла)

[Два не очень разных слова]

Приведена блок-схема алгоритма:



В данной диаграмме: s_1, s_2 – строки из строчных букв английского алфавита; M – длина строки s_1 ; N – длина строки s_2 ; D – матрица размером $M+1$ на $N+1$. Строки матрицы нумеруются от 0, сверху вниз. Столбцы нумеруются от 0, слева направо. Запись $D[i, j]$ означает ячейку на пересечении строки i и столбца j . Символы строк также нумеруются от 0.

В результате выполнения была получена следующая матрица:

```

0 1 2 3 4 5 6
1 1 2 2 3 4 5
2 2 2 3 2 3 4
3 3 3 3 3 2 3
4 4 4 4 3 3 2
5 5 5 5 4 4 3
  
```

6 6 6 5 5 4

7 7 7 7 6 6 5

Определите, какой могла быть строка s_2 , если строка $s_1 = \text{"monocle"}$ (символы кавычек не являются частью строки). В случае, если символ строки s_2 не удаётся определить однозначно, замените его символом *. Например, если в строке из пяти символов первые два определить не удалось, а остальные 3 – это символы **y, m, p**, то в качестве ответа необходимо указать ****ymr**.

6. Телекоммуникационные технологии (1 балл)

[Локальное потепление]

Некоторый сервер с недостаточным охлаждением умеет обрабатывать запросы двух типов. Запрос типа А он обрабатывает за 8 секунд, а запрос типа Б – за 12 секунд. Однако, за каждую секунду, во время которой сервер обрабатывает запрос (вне зависимости от типа запроса), его температура увеличивается на 2 градуса. Если температура сервера достигнет 140 градусов, он необратимо выйдет из строя. Для того чтобы избежать этого, системный администратор решил после обработки каждого запроса (вне зависимости от типа) делать перерыв в X секунд. За каждую секунду перерыва температура сервера снижается на 0.5 градуса. Определите минимально возможное натуральное значение X , при котором сервер никогда не нагреется до 140 градусов, если запросы приходят в следующем порядке: А, А, Б, А, А, Б, А, А, Б и т.д. Температура сервера на момент начала работы – 50 градусов.

7. Технологии обработки информации в электронных таблицах, технологии сортировки и фильтрации данных (2 балла)

[Остаться самим собой]

Дана таблица в режиме отображения формул:

	A	B	C	D	E	F	G
1		2	3	4	5	6	
2		$=IF(MOD(\$A\$1; POW(B\$1; \$A2))=0; DIVIDE(\$A\$1; POW(B\$1; \$A2)); MOD(\$A\$1; POW(B\$1; \$A2)))$					
3		3					
4		4					
5		5					
6		6					
7		7					
8		8					

Формулу из ячейки B2 скопировали во все ячейки диапазона B2:F8. После чего оказалось, что число из ячейки A1 встречается в диапазоне B2:F8 ровно 4 раза. Определите минимальное и максимальное возможное число в ячейке A1, при котором это могло произойти. В ответ запишите сначала меньшее из значений, а затем большее из них.

Примечание: таблица соответствия имён используемых функций.

	Google Sheets	Excel	LibreOffice
Условное вычисление	IF	ЕСЛИ	IF
Возведение в степень	POW	СТЕПЕНЬ	POWER
Остаток от деления	MOD	ОСТАТ	MOD
Частное от целочисленного деления	DIVIDE	ЧАСТНОЕ	QUOTIENT

8. Технологии программирования (3 балла)

[Окна]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	1 секунда
Ограничение по памяти	256 мегабайт

На экране монитора размером $w \times h$ расположены n прямоугольных окон нового текстового редактора. Размер экрана измеряется в символах, то есть текстовая строка длины w символов занимает всю ширину экрана от левого края до правого, а всего на экране могут поместиться друг под другом ровно h таких строк. Занумеруем строки экрана от 1 до h , а столбцы — от 1 до w , и будем обозначать координатами (i, j) позицию символа на пересечении i -й строки и j -го столбца.

Каждое из n окон содержит только текстовое поле, которое по ширине и высоте помещает в себе целое число символов. Окна могут быть разного размера и расположены так, что каждая из $w \times h$ позиций символов на экране принадлежит текстовому полю ровно одного окна. Иными словами, окна не могут накладываться, но могут касаться границами, и в совокупности покрывают всю площадь экрана.

Изначально все текстовые поля в окнах пусты, то есть не содержат никакой текст. Команды вывода строки на экран задаются тройками вида (r, c, s) , означающими, что надо поставить курсор на c -ю слева позицию в r -ю сверху строку монитора и, начиная с этой позиции, напечатать строку s . Символы строки печатаются по очереди слева направо, каждый следующий символ занимает следующую позицию в той же строке и записывается в ней вместо того, что стояло на этой позиции ранее. Так происходит, пока курсор не дойдет до правой границы текущего окна, после чего поведение курсора зависит от *режима*, в котором работает то окно, в котором он находится.

Окна могут работать в трех режимах. При достижении курсором правой границы окна:

В режиме «clip» вывод останавливается, и оставшаяся часть строки просто не выводится.

В режиме «wrap» курсор переносится на первую (ближайшую к левой границе окна) позицию **этого же окна** в следующей строке, после чего вывод продолжается. Если же текущая строка является последней строкой в текущем окне, вывод останавливается;

В режиме «overflow», если сейчас курсор находится в позиции экрана (r, c) , он переносится в позицию $(r, c + 1)$, то есть в следующую позицию на экране, оставаясь в той же строке и игнорируя правую границу окна; после чего вывод продолжается. Если же текущая позиция является последней позицией на экране, то есть $c = w$, курсор переносится на первую позицию на экране в следующей строке $(r + 1, 1)$. Если и строка была последняя, то есть курсор достиг правой-нижней позиции на экране $(r = h \text{ и } c = w)$, вывод останавливается.

Для каждого окна известно, в каком режиме оно изначально работает. Обработайте команды вывода строк и команды изменения режимов окон и выведите состояние экрана после обработки всех команд.

Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке ввода дано единственное целое число t — количество наборов входных данных в тесте ($1 \leq t \leq 50$).

Каждый набор входных данных начинается со строки, в которой даны три целых числа n, w и h — количество окон на экране и размеры экрана, соответственно ($1 \leq w, h; 1 \leq n \leq w \cdot h \leq 2000$).

Следующие n строк набора входных данных описывают окна; i -я строка описывает i -е окно и содержит через пробел четыре целых числа $r_{i,1}, c_{i,1}, r_{i,2}$ и $c_{i,2}$ — номер строки и столбца левого-верхнего угла окна и номер строки и столбца правого-нижнего угла окна ($1 \leq r_{i,1} \leq r_{i,2} \leq h; 1 \leq c_{i,1} \leq c_{i,2} \leq w$), а также строку $mode_i$, равную «clip», «wrap» или «overflow» — изначальный режим, в котором работает i -е окно. Гарантируется, что весь экран покрыт окнами и что каждая позиция на экране принадлежит ровно одному окну.

Затем в отдельной строке следует целое число q — количество команд, которые вам предстоит обработать ($1 \leq q \leq 2000$).

В i -й из следующих q строк дано описание i -й команды в формате

«print $r_i c_i s_i$ », если это команда вывода строки s_i , начиная с позиции (r_i, c_i) ($1 \leq r_i \leq h; 1 \leq c_i \leq w; 1 \leq |s_i|$);

«mode $t_i m_i$ », если это команда изменения режима окна номер t_i на m_i ($1 \leq t_i \leq n; m_i \in \{\text{clip}, \text{wrap}, \text{overflow}\}$).

Гарантируется, что все s_i состоят из маленьких букв латинского алфавита (символы от 'a' до 'z'), и что сумма длин s по всем командам в рамках одного набора входных данных не превосходит 10 000.

Формат выходных данных

Для каждого набора входных данных выведите состояние экрана после обработки всех q команд. Состояние экрана — это h строк по w символов, каждый из которых равен букве на соответствующей позиции экрана, либо '.', если соответствующая позиция пустая и не содержит символ.

Пример

Стандартный ввод	Стандартный вывод
<pre> 2 2 2 2 1 1 2 1 clip 1 2 2 2 overflow 4 print 1 1 abcd print 1 2 xxxx mode 1 wrap print 1 2 yyyy 4 3 3 1 1 1 1 overflow 1 2 1 3 clip 2 1 3 1 wrap 2 2 3 3 wrap 8 print 1 1 abcd mode 2 overflow print 1 2 efgh mode 3 overflow print 2 1 ijkl print 2 3 x mode 4 overflow print 2 3 yzt </pre>	<pre> ay y. aef ijy zt. </pre>

Замечание

Для удобства восприятия наборы входных данных и соответствующий им вывод в примерах в условии отделяются друг от друга пустыми строками. В реальных тестах этих **пустых строк нет!**

9. Технологии программирования (3 балла)

[Делители факториала]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	4 секунды
Ограничение по памяти	256 мегабайт

Математика порой бывает довольно сложной. Однако, когда речь заходит о сложной математике, мало кто может сказать, что тема «факториалы» однозначно попадает в эту категорию. С задачей посчитать факториал целого числа, скорее всего, справится любой участник олимпиады.

Поэтому вам предстоит ответить на немного более сложные запросы. Всего запросов q и i -й запрос задается одним целым числом n_i . Для каждого запроса вам необходимо посчитать $\sigma_0(n_i!)$, то есть количество положительных делителей числа $n_i!$.

Поскольку реальное количество делителей числа $n!$ может быть слишком большим уже даже при $n \approx 100$, посчитайте ответ на каждый запрос по модулю $10^9 + 7$.

Формат входных данных

В первой строке ввода дано целое число q — количество запросов ($1 \leq q \leq 300$).

Во второй строке перечислены q целых чисел n_i — параметры запросов ($1 \leq n_i \leq 10^5$).

Формат выходных данных

Для каждого запроса посчитайте $\sigma_0(n_i!) \bmod (10^9 + 7)$ и выведите в ответ **остаток их суммы по модулю 998 244 353**.

Пример

Стандартный ввод	Стандартный вывод
10 1 2 3 4 5 6 7 10 100 1000	558634260

Замечание

В примере из условия независимые ответы для каждого n_i получаются такие:

$$\sigma_0(1!) = 1;$$

$$\sigma_0(2!) = 2;$$

$$\sigma_0(3!) = 4;$$

$$\sigma_0(4!) = 8;$$

$$\sigma_0(5!) = 16;$$

$$\sigma_0(6!) = 30;$$

$$\sigma_0(7!) = 60;$$

$$\sigma_0(10!) = 270;$$

$$\sigma_0(100!) = 39\,001\,250\,856\,960\,000, \text{ и по модулю } 10^9 + 7 \text{ это равно } 583951250;$$

$$\sigma_0(1000!) \bmod (10^9 + 7) = 972926972, \text{ само значение мы приводить не будем, так как оно слишком большое.}$$