

Задания для 10 класса

Заключительный этап, 10 класс (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (1 балл)

[XYZ₂₀₂₄]

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 (1 балл)

[Цифровой диктофон]

Друг Пети, Павел, пытается сконструировать цифровой диктофон и просит Петю написать прошивку для кодирования и сохранения в памяти оцифрованного аудиосигнала. Петя решил, что будет записывать данные без сжатия и оцифровывать аудиосигнал с частотой дискретизации 88200 Hz, выбрав такую глубину кодирования, чтобы в каждый отсчет времени сохранялось одно из возможных 65536 значений сигнала (для записи значения сигнала в каждый отсчет времени Петя использует минимальное, одинаковое для всех возможных значений количество бит). Поскольку Петя предполагает использовать пару микрофонов, Павел решил записывать звук двухканальным, сохраняя оцифрованный аудиосигнал с указанными параметрами независимо для каждого канала. Опытный Вася обратил внимание Пети на две возможности для уменьшения памяти. Во-первых, можно уменьшить частоту дискретизации в два раза, а во-вторых, записывать с выбранной глубиной кодирования только один канал, а для второго канала записывать для каждого отсчета времени только разность значения сигнала со значением в первом канале, считая, что для этого хватит 1024 возможных значений (для записи разности сигналов в каждый отсчет времени предлагается также использовать минимальное, одинаковое для всех возможных значений разности количество бит). Петя принял оба предложения Васи и обнаружил, что для аудиосигнала длительностью t секунд удалось сэкономить больше 20 МБайт памяти. Определите минимальное **целое** значение t, при котором это возможно. В ответе укажите целое число.

Примечания:

1. При записи оцифрованного сигнала в память не записывается никакая дополнительная информация.
2. 1 МБайт = 2^{20} байт.

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. Кодирование информации. Формальные исполнители (2 балла)

[6 букв]

Есть исходная строка S, состоящая из 6 символов. Результирующая строка R изначально пустая и формируется следующим образом: N раз выполняются следующие два шага:

3. Дописать в конец строки R строку S.
4. Циклически сдвинуть строку S на один символ влево.

Например, для строки S='abcdef' и N=4 получится строка R='abcdefbcdefacdefabdefabc'.

Для некоторой строки S получили результирующую строку R для N=10¹⁰.

Известны некоторые символы в строке R (нумерация символов строки начинается с 1 слева направо):

Номер символа	Значение
10^8+2	a
10^8+4	c
10^8+8	b
10^8+16	d
10^8+3	f
10^8+5	e

Определите и введите в ответ строку S, для которой это возможно.

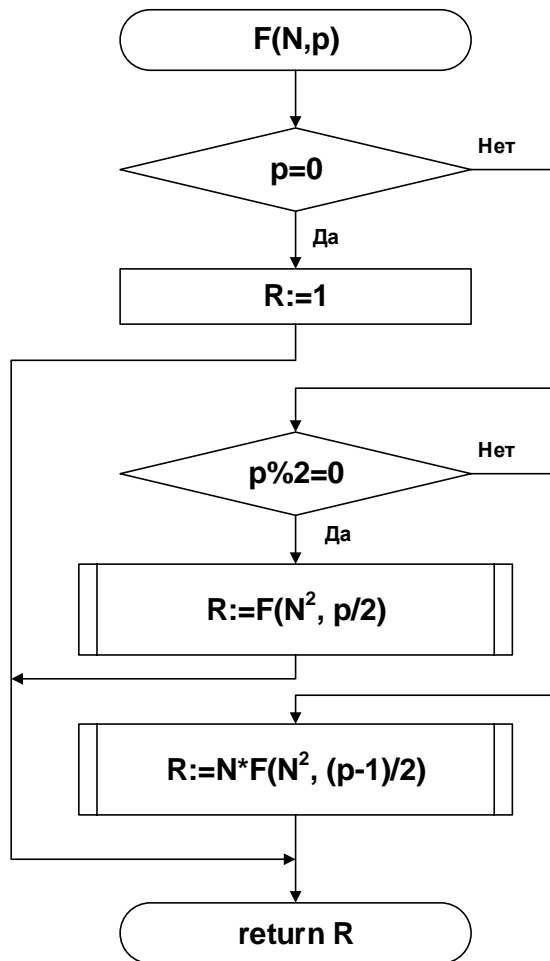
Если вариантов таких строк несколько, введите любую подходящую.

Если такой строки не существует, введите в ответ NULL.

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (2 балла)

[Рекурсия]

Дана блок-схема алгоритма, реализованного в виде рекурсивно вызываемой функции:



Найдите такую пару целых положительных чисел N и p (известно, что $p > 1$), чтобы вызов $F(N, p)$ вернул число 387420489. Если таких пар существует несколько, найдите ту, у которой максимальное значение N . В ответе укажите через пробел сначала значение N и затем значение p . Если такой пары не существует, укажите в ответе NULL.

6. Телекоммуникационные технологии (3 балла).

[Обрыв канала]

В ОС семейства GNU/Linux существует возможность объединения сетевых интерфейсов (сетевых карт) в группы (*bonding*). Ее используют для балансировки нагрузки при передаче и обеспечения отказоустойчивости.

В результате объединения в системе создается виртуальный сетевой интерфейс *bond*. С точки зрения остальных слоев сетевого стека этот интерфейс – обычная сетевая карта, однако при передаче данных через нее реальная передача данных осуществляется через тот или иной физический сетевой интерфейс по выбору ядра операционной системы. Выбор физического интерфейса зависит от режима *bonding*.

Существует 7 режимов *bonding*. Один из них – *balance-xor*. Когда физические интерфейсы объединяются в этом режиме, для выбора физического интерфейса, через который следует передать кадр канального уровня, используются

значения адресов канального уровня (MAC-адресов), содержащиеся в заголовке кадра. Расчет ведется по следующей формуле:

$$(\text{MAC_отправителя} \text{ XOR } \text{MAC_получателя}) \% \text{ число_физических_интерфейсов} = \text{индекс_физического_интерфейса}$$

Здесь XOR – побитовая операция, а % - операция получения остатка от деления. Значения индекса физического интерфейса для передачи начинаются с нуля.

В случае недоступности части физических интерфейсов передача идёт только по активным интерфейсам (то есть изменяется число физических интерфейсов), а индексы самих интерфейсов пересчитываются, сохраняя изначальный порядок в конфигурации и нумерацию с нуля.

Рассмотрим систему с настроенным виртуальным сетевым интерфейсом bond в режиме balance-xor, который включает в себя 4 физических сетевых интерфейса с индексами от 0 до 3. Этой системе поступает несколько кадров для передачи. MAC-адрес отправителя равен 24:01:C7:A3:8B:7E, а MAC-адреса получателей указаны ниже.

00:24:1E:6D:FC:7D
44:45:53:36:A0:88
78:84:3C:94:06:A3
00:22:4C:FA:BE:C9
44:45:53:DC:E6:7A
68:76:4F:9A:99:FF
E0:0C:7F:53:C5:09
44:45:53:7C:70:65
BC:6E:64:3C:AD:EF
A4:5C:27:35:CE:7C
44:45:53:75:DA:1F
9C:5C:F9:74:17:50
00:09:BF:10:AD:FD
44:45:53:74:2A:F4
00:EB:2D:38:D0:F2
00:27:09:00:AC:1B
44:45:53:DC:53:CE
68:76:4F:18:EC:5C
64:B5:C6:58:97:DB
44:45:53:60:96:FE
00:1E:45:9C:9C:FF
E8:DA:20:CF:BB:CB
44:45:53:18:37:68
4C:21:D0:35:31:22
00:1F:C5:A3:8B:43
44:45:53:0B:25:65
00:21:9E:91:53:6B
00:26:59:B4:AE:9A
44:45:53:76:95:14
30:17:C8:8A:84:15

После передачи определённого количества кадров произошёл обрыв сети, из-за чего одновременно стали недоступны два физических интерфейса из четырёх, и оставшиеся кадры передавались по одному из двух активных интерфейсов. Известно, что по интерфейсу с исходным индексом 0 было передано 7 кадров, по интерфейсу с исходным индексом 1 – 12 кадров, по интерфейсу с исходным индексом 2 – 6 кадров, по интерфейсу с исходным индексом 3 – 5 кадров. Определите, какие из интерфейсов стали недоступны и сколько кадров было передано до обрыва канала. В ответе укажите три числа через пробел: номера интерфейсов (0, 1, 2 или 3) в порядке возрастания и количество переданных кадров. Если есть несколько вариантов того, какие интерфейсы могли стать недоступны, выберите любой вариант. Если есть несколько вариантов того, сколько кадров могло быть передано до обрыва, выберите максимальное число.

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							
4							
5							
6							
7							
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 последней, 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_i 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
2	ay
2 2 2	y.
1 1 2 1 clip	aef
1 2 2 2 overflow	ijy
4	zt.
print 1 1 abcd	
print 1 2 xxxx	
mode 1 wrap	
print 1 2 yuyu	
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	

Стандартный ввод	Стандартный вывод
print 2 1 ijkl	
print 2 3 x	
mode 4 overflow	
print 2 3 yzt	

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)

[Скрещивание]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	3 секунды
Ограничение по памяти	256 мегабайт

Вы — владелец зоопарка, в котором сейчас обитают n редких экзотических видов змей, i -й из которых имеет опасность a_i .

Держать животных в неволе вам не очень хочется, но и выпускать опасных змей в общее пространство тоже, разумеется, нельзя. Для исправления этой проблемы вы решили скрестить некоторые виды, от чего их экзотичность меньше не станет, а вот опасность может уменьшиться. Для одного скрещивания вы можете выбрать два вида змей под номерами i и j , и скрестить их в новый вид, имеющий опасность $a_{i,j}^* = a_i \oplus a_j$, где за \oplus обозначена операция побитового исключающего «ИЛИ» (также обозначается как хог или \wedge).

Чтобы не получать слишком похожие виды, каждый из имеющихся n видов может участвовать только в одном скрещивании. Также невозможно скрестить два вида, хотя бы один из которых уже является результатом скрещивания каких-то из изначальных n видов. Иными словами, скрещивать можно только исходные виды, и только по парам.

В конце вы получаете новый набор видов, в который войдут исходные виды, не поучаствовавшие в скрещиваниях, а также все результаты скрещиваний. То есть исходные виды, которые были скрещены с какими-то еще, в этот набор не войдут (опять же, потому что нет смысла отводить в зоопарке отдельное место под несколько близких видов — посетители все равно не увидят разницу).

Определите, какие пары видов змей следует скрестить, чтобы максимум из опасностей полученного набора видов был как можно меньше.

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 ($1 \leq t \leq 50$). Далее следуют сами наборы входных данных.

В первой строке набора входных дано целое число n — количество видов экзотических змей в наличии изначально ($1 \leq n \leq 50\,000$). Гарантируется, что сумма n по всем наборам входных данных не превосходит 10^5 .

Во второй строке перечислены n целых чисел a_i — значения опасности этих видов ($0 \leq a_i \leq 10^9$).

Формат выходных данных

Для каждого набора входных данных выведите в отдельной строке единственное целое число — минимальное возможное значение максимальной опасности, которое можно получить такими скрещиваниями.

Пример

Стандартный ввод	Стандартный вывод
4	1
3	21
1 2 3	4
7	8
1 4 9 16 25 36 49	
6	
0 1 2 5 6 7	
10	
5 7 14 10 12 2 1 13 3 11	

Замечание

В первом примере выгодно скрестить 2 и 3, и получить исходный вид с $a_1 = 1$ и новый с $a_{2,3} = 1$.

Во втором примере скрещиваются 16 с 25 (получается $a_{4,5} = 9$) и 36 с 49 (получается $a_{6,7} = 21$).

В третьем примере скрещиваются 2 с 6 и 5 с 7.