A. Excel

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

MicroSoft Excel – дорогая программа, Василию такая программа не по карману, но он с этой программой знаком по пробным версиям. Василий решил написать свой VasyaSoft Excel для бухгалтерских расчётов в биткоинах. Он просит вас за время соревнования реализовать одну из ожидаемых в программе возможностей.

Вам дано поле размером до 8 на 8 ячеек, для прототипа больше пока не требуется. В каждой ячейке имеются вещественные числа с точно восемью знаками после запятой (минимальная 10^{-8} часть от биткоина – сатоши). В некоторых имеется операция сложения двух значений, каждое значение в этом случае представляет собой ссылку на ячейку. Входные и итоговые значения не могут быть больше 21×10^6 биткоинов.

	A	B
1	0.00000001	9.99999999
2	=A1+B1	=B4+B3
3		=B4+B4
4	=B2+A1	=A1+A3

	A	B
1	0.00000001	9.99999999
2	10.00000000	0.00000003
3		0.00000002
4	0.00000004	0.00000001

Входные данные

Таблица VasyaSoft Excel (формат см. в примерах). Отрицательные числа не используются, для этого в бухгалтерии принято одни значения относить к дебету, другие к кредиту. Имена столбцов именуются слева направо последовательно ABCDEFGH. Строки нумеруются сверху вниз последовательно 12345678. Формулы выравнены по левому краю ячеек. Значения в пустых ячейках (содержат ноль и более пробелов) считать равными нулю.

Выходные данные

Выведите результирующую таблицу в том же формате, что и во входных данных. Учтите, что ширина ячеек в каждом столбце должна быть минимально возможна и в каждой строке один и тот же столбец имеет одинаковую ширину. Выравнивание значений должно быть по правому краю ячеек. Пустые ячейки в выходном файле должны остаться пустыми (содержать пробелы при выводе). Для выравнивания используйте пробелы. Если вычислить значение в любой из ячеек невозможно - выведите «ERROR».

Примеры

Ввод	Вывод
+----------+----------+ |0.00000001|9.99999999| +----------+----------+ |=A1+B1 |=B4+B3 | +----------+----------+ | |=B4+B4 | +----------+----------+ |=B2+A1 |=A1+A3 | +----------+----------+	+-----------+----------+ | 0.00000001|9.99999999| +-----------+----------+ |10.00000000|0.00000003| +-----------+----------+ | |0.00000002| +-----------+----------+ | 0.00000004|0.00000001| +-----------+----------+

Ввод	Вывод
+----------+----------+ |=A2+B1 |=A1+A1 | +----------+----------+ |0.99999999|0.00000001| +----------+----------+	ERROR
+------++ |=B1+B1|| +------++	+----------++ |0.00000000|| +----------++

B. Форматирование

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Разные люди по-разному записывают номер мобильного телефона, например, один и тот же номер может быть записан следующими способами: +3 (141) 592-65-35, так +31415926535, +3 141 592 65 35, +3 (141) 592 65 35, 3141 59-26-535 и т.д.

Василий просит вас создать программу, которая получая на вход записанные в различных форматах номера телефонов состоящие из 11 цифр, приводит их к одному формату.

В формате могут использоваться следующие знаки: -+()*:.# и, конечно, пробел. В последовательности формата всегда 11 знаков #, каждый такой знак должен быть заменён на очередную цифру номера.

Входные данные

Первая строка содержит формат вывода номеров телефона. Далее идут не более 100 номеров телефонов в различных форматах представления. Каждый номер расположен на отдельной строке и может содержать знаки из условия задачи. Гарантируется, что ни формат, ни один из номеров не начинается и не заканчивается пробелом, каждый номер в точности содержит 11 цифр номера. Каждая строка входного файла короче 100 символов.

Выходные данные

Первая строка – формат вывода номера телефона. Далее все номера входного файла в формате, соответствующем первой строке входного файла.

Пример

Ввод	Вывод
+# (###) ###-##-## 31415926536 +2 (7) 18-28-18-28--5 *1 41 :) 42-135-62--4 :(	+# (###) ###-##-## +3 (141) 592-65-36 +2 (718) 281-82-85 +1 (414) 213-56-24

C. Алгоритм Луна

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Василий, видя, как повсюду внедряются пластиковые карты, решил разобраться, как там всё устроено. Сейчас он разбирается с алгоритмом Лу́на, который используется для вычисления контрольной суммы номера пластиковой карты в соответствии со стандартом ISO/IEC 7812.

Оригинальный алгоритм проверки контрольной суммы, описанный разработчиком:

1. Цифры проверяемой последовательности нумеруются справа налево.
2. Цифры, оказавшиеся на нечётных местах, остаются без изменений.
3. Цифры, стоящие на чётных местах, умножаются на 2.
4. Если в результате такого умножения возникает число больше 9, оно заменяется суммой цифр получившегося произведения – однозначным числом, то есть цифрой.
5. Все полученные в результате преобразования цифры складываются. Если сумма кратна 10, то исходные данные верны.

Входные данные

Последовательность цифр не более 20, одна из цифр заменена на знак вопроса. Место расположения знака вопроса – место расположения контрольной цифры.

Выходные данные

Найдите значение контрольной цифры, подставьте вместо знака вопроса и выведите получившуюся последовательность цифр.

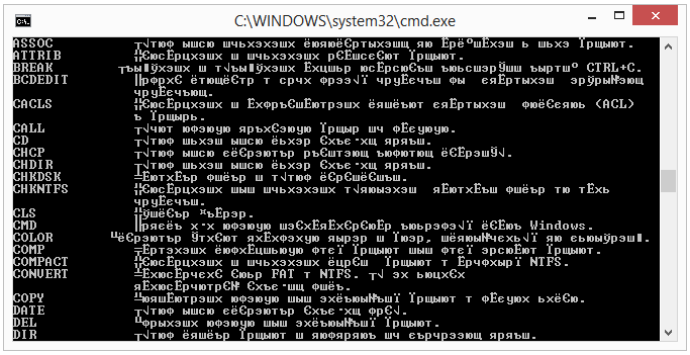
Примеры

Ввод	Вывод
456126121234546?	4561261212345467

D. DOS

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

К Василию обратился его друг Сергей, которому попались документы, созданные ещё во времена популярности DOS. Во времена популярности DOS – дисковой операционной системы использовалась кодировка русских букв, которая сейчас в windows имеет название cp866.



У Василия есть текстовый редактор, который использует кодировку cp1251.

В cp866 для кодирования русских букв используются следующие коды (даны в шестнадцатеричном виде): от А до Я (без Ё) коды с 80 до 9F, Ё – F0, от а до п (без ё) коды с A0 до AF, от р до я коды с E0 до EF, ё – F1.

В cp1251 для кодирования русских букв используются следующие коды (даны в шестнадцатеричном виде): от А до Я (без Ё) коды с C0 до DF, Ё – A8, от а до я (без ё) коды с E0 до FF, ё – B8.

Напишите программу, которая перекодирует файл из cp866 в cp1251.

Коды символов: пробел, запятая, точка, двоеточие, точка с запятой, восклицательный и вопросительные знаки, круглые, квадратные, фигурные и угловые скобки, цифры и буквы латинского алфавита, арифметические знаки в обеих кодировках совпадают.

Входные данные

Текст, который может содержать только символы, перечисленные в абзаце начинающегося с «Коды символов: …», и символы русского алфавита в кодировке cp866. Во входном файле не более 10000 символов.

Выходные данные

Исходный текст, перекодированный в cp1251.

Примеры

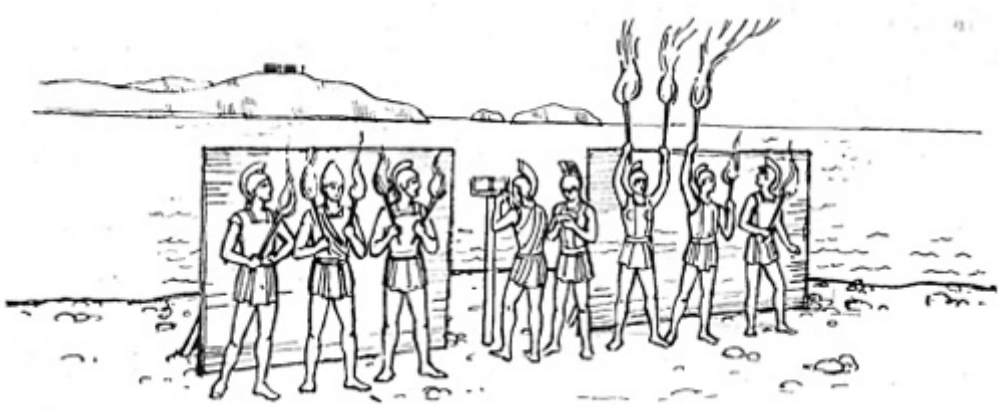
Ввод (кодировка cp866)	Вывод (кодировка cp1251)
Привет, Мир!	Привет, Мир!
Это равенство 2*2=4-4 верно?	Это равенство 2*2=4-4 верно?
First - первый Second - второй	First - первый Second - второй

E. Античный телеграф

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Василий не успевает заниматься всем, чем ему интересно. Он узнал, что есть системы, в которых логическая единица кодируется повышенным потенциалом, а есть и такие, где пониженным. Инженеры разрабатываемых систем делают тот или иной выбор в результате рассмотрения готовых решений и их цен, энергоэффективности того или иного кодирования, с учётом помехоустойчивости и других параметров.

Василий, прочитав про различные используемые способы передачи информации в античные времена, сделал робота, который использует для передачи информации огонь. Робот по его задумке будет передавать числа как двоичные 32-битные последовательности от младшего бита к старшему. Каждый бит последовательности он передаёт путём зажигания «факела» (зажигалки) на разное время (хорошо отличимое человеком).



Он понимает, что для экономии газа в зажигалке для чисел содержащих много единиц в своей двоичной записи, нужно единицы кодировать малым временем горения огня, а нулей более долгим, а для чисел, содержащим нулей больше чем единиц наоборот. Для передачи формы кодирования, он придумал специальный код. Чтобы часто не менять способ кодирования (на это тоже уходит горючая жидкость зажигалки) он просит вас написать программу. Программа должна отсортировать все необходимые для передачи числа в следующем порядке: сначала идут числа, в двоичной записи которых количество единиц больше, затем меньше, если количество единиц одинаковое, то сначала идут меньшие по значению числа (в дальнейшем он собирается усовершенствовать передачу, передавать только разницу между соседними значениями, это позволит сжать информацию).

Входные данные

Через пробел до 10^5 целых неотрицательных чисел по величине меньших 10^9.

Выходные данные

Через пробел числа из входных данных, упорядоченные в соответствии с условием задачи.

Примеры

Ввод	Вывод
5 3 1 7 2	7 3 5 1 2

Примечание:

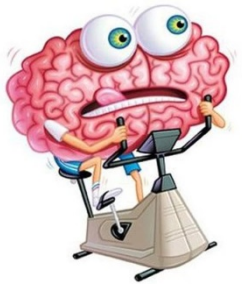
Двоичный код чисел из примера выглядит так (даны младшие 4 двоичных разряда, остальные старшие 28 равны нулям): 7 – 0111, 3 – 0011, 5 – 0101, 1 – 0001, 2 – 0010.

Семь идёт раньше всех, так как содержит наибольшее количество единиц в двоичной записи, тройка имеет две единицы также как и пятёрка, но имеет меньшую величину (3<5). Единица и двойка имеют наименьшее количество единиц – по одной, но так как один меньше 2, единица идёт раньше двойки.

F. Разминка

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Разминка подготавливает организм к более интенсивным нагрузкам. Выполнение разминки может предохранять спортсмена от травм и является важной частью тренировки. Эта задача позволит «разогреть» мозг и не травмировать его логико-вычислительными процессами в ходе соревнования (мы на это надеемся).



Входные данные

Через пробел три целых числа: a, b, c ($-10^5 \le a, b, c \le 10^5$).

Выходные данные

Один из арифметических знаков: «+», «–», «*» или «/», такой, чтобы выражение a X b = c, где X – выведенный программой знак, было верным. Если получить результат невозможно, программа должна вывести impossible. Если вариантов несколько, то программа должна вывести more.

Пример

Ввод	Вывод
1 2 3	+
0 1 0	more
1 0 2	impossible
-1 0 0	*
12 4 3	/
4 -3 7	-

G. Ремонт

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

«Папа у Васи силён в математике!» – неоднократно озвученные слова песни в одной знаменитой комедии снятой на киностудии «Ленфильм» в 1973 году.

Но и Василий не слаб. Намечается ремонт в его комнате, нужно закупить обои под покраску. Василий всё обмерил, размеры прямоугольного помещения M на N и высоту H, количество проёмов (окон и дверей) и их размеры X_i и Y_i, через интернет магазины узнал длину трубок L и их ширину W (клеят обои не внахлёст). Василий всё посчитал, но понимает, что такие расчёты не каждый может сделать, но всем вручную не посчитаешь. Нужно реализовать веб-сервис для автоматизации таких расчётов.

Пока Василий занимается оформлением сайта, напишите программу, которая будет по известным данным находить минимально необходимое целое количество трубок обоев. Поклейка обоев будет всех стен на всю высоту.

Входные данные

Первая строка содержит три вещественных положительных числа M, N, H (все числа меньше 15) – размеры помещения в метрах с точностью до десятых. Затем на отдельной строке одно целое положительное число K (меньшее 10) – количество проёмов в помещении. Далее идёт K строк с целочисленными размерами X_i и Y_i (в сантиметрах), размеры проёмов не могут быть больше размеров стен. Затем на отдельной строке указаны два числа: L (в метрах с точностью до десятых), W (целое количество сантиметров). Длина рулонов не может быть больше 20 метров, ширина не более 1,5 метров.

Выходные данные

Одно целое число – минимальное целое количество рулонов для оклейки всех стен.

Пример

Ввод	Вывод
2.2 4.2 2.5 2 200 70 147 142 10 106	3

H. Электроника МК-52

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

Папа Василия в молодости тоже чуть-чуть занимался программированием. У него был и сейчас есть программируемый микрокалькулятор «Электроника МК-52» с обратной польской записью для проведения инженерных расчётов (см. рисунок).

Обратная польская запись – форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Например, для вычисления такого выражения (6+8)×3= в обратной польской записи необходимо записать так 3 6 8 + ×. Заметьте, знака равно при этом нет!

Вот что рассказал Василию его папа: «У микроконтроллера имеется стек регистров: X, Y, Z и T. Когда вводят новое число оно попадает в регистр X. При нажатии на кнопку B↑ данные, хранящиеся до нажатия в регистре T теряются, в этот регистр записываются данные, хранящиеся в регистре Z, в регистр Z записываются данные, хранящиеся в Y, в Y записываются данные из X, в X остаются данные которые были введены, но при вводе новых данных они перепишут данные в регистре X. После нажатия одной из кнопок, указывающих на арифметические действия (+×–÷), арифметическое действие производится над операндами, хранящимися в регистрах X и Y, результат помещается в регистр X, содержание регистра Z переписывается в регистр Y, значение в T переписывается в Z».

Входные данные

Дана строка, где указаны действия пользователя, кнопка B↑ обозначена знаком ^, арифметические действия знаками +*-/. Отрицательные величины не вводятся, но могут получатся в результате вычисления. Гарантируется, что вводимая строка имеет длину не более 100 знаков, и во всех промежуточных и итоговых результатах вы имеете действие с целыми числами по абсолютному значению меньшими 1000. В начальный момент в регистрах X, Y, Z и T записаны нули. Все операции выполнимы.

Выходные данные

4 числа через пробел – значения регистров X, Y, Z и T.

Пример

Ввод	Вывод
7^5^1^2-+*14/	2 7 7 7

Примечание:

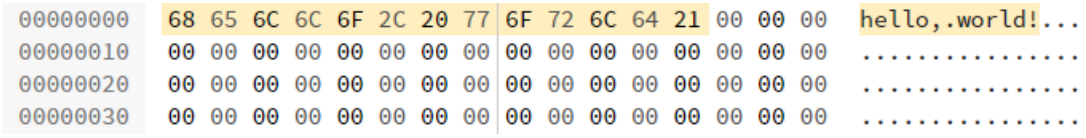
В примере вычисляется следующее выражение (7×(5+(1–2)))/14. В таблице ниже показано как менялись состояния регистров после очередного нажатия кнопки пользователем (данные из примера).

регистры		Действия пользователя												
		7	^	5	^	1	^	2	-	+	*	1	4	/
T	0	0	0	0	0	0	7	7	7	7	7	7	7	7
Z	0	0	0	0	7	7	5	5	7	7	7	7	7	7
Y	0	0	7	7	5	5	1	1	5	7	7	28	28	7
X	0	7	7	5	5	1	1	2	-1	4	28	1	14	2

I. HEX-редактор

Ограничение по времени: **1000 миллисекунд**
Ограничение по памяти: **65000 кибибайт**

«Каждый уважающий себя программист должен написать свой hex-редактор» - так думает Василий, и потому, не тратя время сел за его написание. Пока он разрабатывает дизайн, вам предлагается реализовать одну из функций редактора – отображение текстовой информации по кодам символов. Коды символов могут быть в диапазоне от 0 до 255 (в шестнадцатеричном виде от 00 до FF).



Входные данные

Строка шестнадцатеричных цифр без пробелов и каких-либо других символов. Строка всегда состоит из чётного количества шестнадцатеричных цифр и её длина не более 128 символов.

Выходные данные

Массив значений из 4 строк по 16 символов в каждой. Коды символов от 33 до 127 отображаются как есть, другие значения всегда точкой. При отсутствии данных, коды следует считать равными нулю.

Пример

Ввод	Вывод
68656C6C6F2C20776F726C6421	hello,.world!.. ....