

Во всех задачах вывод вашей программы будет проверяться с правильным ответом побайтово, то есть никакие иные символы, кроме приведенных в примерах (пробелы, символы новой строки и т.д.) выводиться не должны.

Задача А. Hello, world!

Напишите программу, которая выводит на стандартный вывод строку `Hello, world!`

Пример

stdin	stdout
	Hello, world!

Задача В. Простой цикл

Программа получает на вход один байт, принимающий значения от 0 до 255. Программа должна вывести несколько символов “#”, количество выведенных символов должно быть равно значению считанного байта.

Пример

stdin	stdout
0	##### #####

В приведенном примере программа должна вывести 48 символов “#”.

Задача С. Алфавит

Программа получает на вход два байта, принимающие значения от 0 до 255, значение второго байта не меньше, чем первого байта. Программа должна вывести все ASCII-символы, коды которых принимают значения от первого считанного байта до второго (включительно).

Пример

stdin	stdout
AZ	ABCDEFGHIJKLMNQRSTUWXYZ

Задача D. Учимся считать

Программа получает на вход последовательность, содержащую от 1 до 254 символов “\$”, а затем один символ “#”. Выведите один байт, ASCII-код которого равен количеству считанных символов “\$”.

Пример

stdin	stdout
\$ \$#	1

В приведенном примере 49 символов “\$”.

Задача Е. Сложение

Программа получает на вход два байта, принимающие значения от 0 до 255. Программа должна вывести один байт, ASCII-код которого равен сумме ASCII-кодов считанных байтов. Гарантируется, что эта сумма не превосходит 255.

Пример

stdin	stdout
01	a

Задача F. Умножение

Программа получает на вход два байта, принимающие значения от 0 до 255. Программа должна вывести один байт, ASCII-код которого равен остатку от деления произведения ASCII-кодов считанных байтов на 256.

Пример

stdin	stdout
BC	F

В примере вычислено значение $66 \times 67 \equiv 70 \pmod{256}$.

Задача G. Возведение в степень

Программа получает на вход два байта a и b , принимающие значения от 1 до 255. Программа должна вывести один байт, ASCII-код которого равен остатку от деления значения a^b на 256.

Пример

stdin	stdout
31	s

В примере вычислено значение $51^{49} \equiv 115 \pmod{256}$.

Задача H. Название цифры – easy

Программа получает на вход один символ, равный 0 или 1 и должна вывести одно из двух слов в зависимости от ее значения: `zero` или `one`.

Пример

stdin	stdout
0	zero
1	one

Задача I. Название цифры – hard

Программа получает на вход один символ от 0 до 9 и должна вывести одно из слов в зависимости от ее значения: `zero`, `one`, `two`, `three`, `four`, `five`, `six`, `seven`, `eight`, `nine`.

Пример

stdin	stdout
0	zero
9	nine

Задача J. Логическое И

Программа получает на вход два символа, каждый из которых принимает значение “0” или “1” и должна вывести значение логической функции “И” для данных символов.

Пример

stdin	stdout
00	0
01	0
10	0
11	1

Задача K. Логическое ИЛИ

Программа получает на вход два символа, каждый из которых принимает значение “0” или “1” и должна вывести значение логической функции “ИЛИ” для данных символов.

Пример

stdin	stdout
00	0
01	1
10	1
11	1

Задача L. Логическое Исключающее ИЛИ

Программа получает на вход два символа, каждый из которых принимает значение “0” или “1” и должна вывести значение логической функции “Исключающее ИЛИ” для данных символов.

Пример

stdin	stdout
00	0
01	1
10	1
11	0

Задача M. Голосование

Программа получает на вход три символа, каждый из которых принимает значение “0” или “1” и должна вывести тот из введенных символов, который встречается чаще.

Пример

stdin	stdout
101	1
000	0

Задача N. Частное

Программа получает на вход два байта a и b ($0 \leq a < 256$, $0 < b < 256$). Программа должна вывести один байт, ASCII-код которого равен целой части частного a и b ($[a/b]$).

Пример

stdin	stdout
z!	0x03

В примере вычислено значение $[122/33] = 3$. Выходной файл содержит один байт с ASCII-кодом 03₁₆.

Задача O. Остаток

Программа получает на вход два байта a и b ($0 \leq a < 256$, $0 < b < 256$). Программа должна вывести один байт, ASCII-код которого равен остатку от деления a на b .

Пример

stdin	stdout
q@	1

В примере вычислено значение $113 \bmod 64 = 49$.

Задача P. Максимум

Программа получает на вход два символа и должна вывести один символ из двух введенных — тот, ASCII-код которого больше.

Пример

stdin	stdout
01	1
aA	a

Задача Q. Сортировка двух элементов

Программа получает на вход два символа, ASCII-коды которых лежат в интервале $[33, 127]$. Выведите эти символы в порядке возрастания их ASCII-кодов.

Пример

stdin	stdout
01	01
aA	Aa

Задача R. Сортировка трех элементов

Программа получает на вход три символа, ASCII-коды которых лежат в интервале [33, 127]. Выведите эти символы в порядке возрастания их ASCII-кодов.

Пример

stdin	stdout
SIS	ISS

Задача S. Сложение с переносом

Программа получает на вход два байта, ASCII-коды которых соответствуют цифрам. Вычислите их сумму и выведите в виде ASCII-символов, соответствующих десятичной записи этой суммы.

Пример

stdin	stdout
45	9
89	17

Задача T. Алгоритм Евклида

Программа получает на вход два байта, не равных одновременно 0. Выведите байт, ASCII-код которого равен наибольшему общему делителю ASCII-кодов исходных символов.

Пример

stdin	stdout
iF	#

Пример иллюстрирует тот факт, что $\text{НОД}(105,70)=35$.

Задача U. Проверка на простоту

Дан один символ, ASCII-код которого принимает значения от 2 до 200. Выведите символ "1", если этот код является простым числом или выведите символ "0", если число — составное.

Пример

stdin	stdout
C	1
D	0

Пример иллюстрирует тот факт, что 67 — простое число, а 68 — составное.

Задача V. Сортировка двадцати элементов

Программа получает на вход двадцать символов, ASCII-коды которых лежат в интервале [33, 127]. Выведите эти символы в порядке возрастания их ASCII-кодов.

Пример

stdin	stdout
qwertyuiopasdfghjklz	abcdefghijklmnopqrstuwyz

Задача W. Обращение строки

Необходимо вывести данную строку в обратном порядке.

Первый байт входных данных содержит количество символов в строке k (от 0 до 255). Далее идет k байт, принимающих значения от 0 до 255. Необходимо вывести данные k байт в обратном порядке.

Пример

stdin	stdout
!abcdefghijklmnopqrstuvwxyzABCDEFGH	GFEDCBAzyxwvutsrqponmlkjihgfedcba

Задача X. Массив

Массивом называется структура данных, позволяющая обращаться к элементу по его индексу, то есть поддерживающая следующие операции:

Set(k, v) — записывающая значение v в элемент массива с индексом k .

Get(k) — возвращающая значение элемента массива с индексом k .

Реализуйте массив байтов с индексами [0..255] с указанными операциями. Первоначально массив заполнен нулями.

Формат входного файла

Первый байт входных данных содержит число N — количество операций типа **Set** для данного массива ($0 \leq N \leq 20$). Далее идет $2N$ байт, содержащих описание N операций. Каждая операция **Set** задается двумя байтами: индексом элемента k и значением v , которое необходимо записать в k -й элемент массива.

Следующий байт входных данных содержит число M — количество операций типа **Get** для данного массива ($0 \leq M \leq 20$). Далее идет M байт, описывающих операции **Get**. Каждая операция задается одним байтом — индексом элемента k .

Формат выходного файла

Программа должна вывести M байт — для каждой операции типа **Get** необходимо вывести значение соответствующего элемента массива.

Пример

stdin	stdout
0x04	@CB
0A	
1B	
0C	
A@	
0x03	
A01	

В примере вставлены фиктивные концы строк для удобства чтения и используется запись 0x03 и 0x04 для обозначения байтов с соответствующими шестнадцатеричными кодами.