

## Задача 5. Побитовый вывод

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Есть алфавит (набор) из  $N$  различных символов, пронумерованных числами от 0 до  $N-1$ . Есть текст из  $M$  символов, все символы которого выбраны из этого набора. Для каждого из  $N$  символов задано, какой последовательностью битов он кодируется. Нужно перекодировать весь текст в битовый массив и записать битовый массив в файл.

В данной задаче все числа записаны с little-endian порядком байтов. Более того, при записи битового массива мы считаем, что биты внутри байта также идут в little-endian порядке.

### Формат входных данных

В первых 4 байтах входного файла дано целое число  $N$  — количество различных символов в алфавите ( $1 \leq N \leq 10^4$ ). Далее для каждого из этих символов идёт описание битовой последовательности, на которую его надо заменять. Описание начинается с целого 4-байтового числа  $L$  — количества битов в последовательности ( $1 \leq L \leq 64$ ). Далее идёт  $L$  байтов, каждый байт принимает значение 0 или 1 и по сути задаёт один бит в последовательности.

Описания битовых последовательностей даны в порядке увеличения номера символа. Биты в каждой последовательности даны в том порядке, в котором их надо записывать в выходной битовый массив.

Далее в файле записано 4-байтовое целое число  $M$  — длина текста в символах ( $1 \leq M \leq 10^6$ ). Наконец, в конце файла записан текст как  $M$  символов, каждый записан как 2-байтовое целое число. Гарантируется, что все символы лежат в диапазоне от 0 до  $N-1$ .

### Формат выходных данных

Поскольку записать в файл можно только целое количество байтов, битовый массив дополняется минимальным количеством нулевых битов так, чтобы его длина делилась на 8.

### Пример

input.txt															
04	00	00	00	0D	00	00	00	00	01	01	00	01	00	01	01
00	00	01	01	01	0D	00	00	00	00	01	01	01	01	01	00
00	01	01	00	00	01	08	00	00	00	01	01	01	00	01	00
00	01	06	00	00	00	01	01	01	01	00	01	08	00	00	00
03	00	01	00	00	00	01	00	00	00	01	00	00	00	01	00
output.txt															
AF	CF	5C	E3	3E	73	8D	FB	CC	35	EE	33	01			

### Пояснение к примеру

Пример в бинарном виде можно скачать по ссылке.

В алфавите 4 символа. Символ 0 заменяется на 1101011000111, символ 1 на 0111110011001, символ 2 на 11101001, а символ 3 на 111101. В тексте 8 символов: 3, 1, 0, 1, 0, 1, 0, 1.

В результате кодирования получается битовый массив: 11110101 11110011 00111010 11000111 01111100 11001110 10110001 11011111 00110011 10101100 01110111 11001100 1. В последней группе только один единичный бит, он дополняется нулями до единичного байта.