

NSU-2023-T01L3e01

Исключающее ИЛИ (XOR) всех битов в битовой строке играет важную роль в цифровых коммуникациях и технологиях хранения данных. Рассматривая байт X , размещенный по адресу x , как 8-битовую строку, вычислите XOR всех битов. Установите переменную-результат res в 0 или 1 в соответствии со значением XOR всех битов X .

Совет: сдвиньте X семь раз и используйте `addc` чтобы добавить бит переноса к регистру, затем очистите все биты в этом регистре, кроме младшего. Этот метод основан на том, что XOR – это битовое сложение по модулю 2. «Модуль 2», или остаток от деления на 2, для двоичных чисел равен младшему биту. XOR всех битов иногда называют *битом четности*, потому что он равен 0, если в исходной строке было нечетное число единиц, и 1 – если четное. Действительно, если добавить этот бит к строке, то общее число единиц в ней всегда будет четным.

Или используйте команду Cdm-8 `xor` и сдвиги. Или любую другую магию, которая работает.

Тестер сообщит вам, сколько команд он исполнил в вашей программе до ее остановки, а также сколько команд потребовалось референтному решению для вычисления результата. Посмотрите, насколько быстра ваша программа по сравнению с референтной. Вы можете отправлять ваше решение несколько раз, даже после того, как оно было засчитано.

Заполните шаблон решения, предоставленный в `NSU-2023-T01L3e01.asm`, в котором мы объявили метки `x` и `res`.

Соглашение о вызовах

Оставьте адрес результата в `r0`. (Это уже закодировано в шаблоне).