

NSU-2023-T01L3e02

Напишите программу, которая получает два 16-битных беззнаковых числа A и B , размещенные в памяти в прямом (little-endian, от младшего к старшему) порядке, вычисляет $A - 2B$ и сохраняет результат как 16-битное (2-байтовое) число, также в порядке от младшего к старшему. Предполагайте, что результат не может быть отрицательным.

Напоминание: 2-байтовое число, сохраненное в порядке от младшего к старшему начиная с адреса памяти X , должно иметь младший байт (правую сторону в арабской позиционной записи) по адресу X и старший байт (левую сторону) по адресу $X + 1$.

Начальные данные вашей программы должны быть размещены в памяти следующим образом:

location	label	description	type
INPUTS+0	<i>aLo</i>	младший байт A	8-битная строка
INPUTS+1	<i>aHi</i>	старший байт A	8-битная строка
INPUTS+2	<i>bLo</i>	младший байт B	8-битная строка
INPUTS+3	<i>bHi</i>	старший байт B	8-битная строка

Проверьте свою программу, размещая разные пары двухбайтовых значений, размещая соответствующие битовые строки по адресам памяти *aLo*, *aHi*, *bLo* и *bHi*.

Заполните шаблон решения, предоставленный в `NSU-2023-T01L3e02.asm`, в котором мы объявили метки *aLo*, *aHi*, *bLo*, *bHi*, *resLo* и *resHi*.

Соглашение о вызовах

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