

Problem 01-1

На языке ассемблера напишите подпрограмму `satsum`, которая принимает два 32-битных беззнаковых целых числа и возвращает 32-битное целое беззнаковое число. Подпрограмма выполняет сложение с насыщением двух беззнаковых чисел. То есть, если сложение вызывает переполнение, результатом будет максимальное целое 32-битное беззнаковое число.

Подпрограмма должна соблюдать стандартные соглашения о вызовах. Подпрограмма должна использовать стандартные прологи и эпилоги. В случае архитектуры `i686` для доступа к параметрам используйте регистр стекового кадра.

Problem 01-2

На языке ассемблера ARM напишите функцию `main`, которая будет выводить на стандартный поток вывода значение глобальной переменной `varspec` типа `unsigned int`.

Переменная `varspec` будет объявлена в секции `.data` в другом файле, с которым ваша программа будет компилироваться. В вашей программе объявления переменной `varspec` быть не должно.

Функция `main` должна завершаться с кодом завершения 0.

Для вывода значения переменной используйте `printf` со спецификацией формата `%u`

Problem 01-3

На языке ассемблера ARM напишите функцию `main`.

На стандартном потоке ввода подается одно целое число (типа `int`).

На стандартный поток вывода напечатайте это число, взятое с обратным знаком. После вывода числа выведите символ перехода на новую строку.

Функция `main` должна завершаться с кодом завершения 0.

Problem 01-4

На языке ассемблера ARM напишите функцию `main`.

На стандартном потоке ввода подаются два числа (типа `int`).

На стандартный поток вывода напечатайте "исключающее или" этих чисел. После вывода числа выведите символ перехода на новую строку.

Функция `main` должна завершаться с кодом завершения 0. Для чтения двух чисел используйте один вызов `scanf`.