

Отчёт по ИДЗ №1

ФИО: Сайфутдинов Рафаэль Рустамович

Группа: БПИ216

Вариант задания: 8

Условие задания:

8. Сформировать массив В по следующим правилам:

- если $A_i > 5$, то увеличить элемент на 5,
- если $A_i < -5$, то уменьшить на 5,
- остальное обнулить.

Тесты, демонстрирующие проверку программ:

Подготовлено 5 тестов, покрывающих достаточное количество возможных результатов работы программы, в том числе используя отрицательные значения в качестве размера массива или значения более 10.

Первым числом вводится N – количество элементов в массиве. Затем следуют N чисел A_i – элементы массива A.

Тест №1:

Команда: `./main.exe < Tests/test1.in`

Входные данные: 5 1 2 3 4 5

Ожидаемые выходные данные: 0 0 0 0 0

Фактические выходные данные: 0 0 0 0 0

Тест №2:

Команда: `./main.exe < Tests/test2.in`

Входные данные: 5 -55 -5 0 5 15

Ожидаемые выходные данные: -60 0 0 0 20

Фактические выходные данные: -60 0 0 0 20

Тест №3:

Команда: `./main.exe < Tests/test3.in`

Входные данные: 0 1 2 -3

Ожидаемые выходные данные:

Фактические выходные данные:

Тест №4:

Команда: `./main.exe < Tests/test4.in`

Входные данные: -5

Ожидаемые выходные данные: Ошибка: размер массива должен быть в диапазоне 0 <= n <= 10. Запустите программу еще раз!

Фактические выходные данные: Ошибка: размер массива должен быть в диапазоне 0 <= n <= 10. Запустите программу еще раз!

Тест №5:

Команда: ./main.exe < Tests/test5.in

Входные данные: 11 1 2 3 4 5 6 7 8 9 10 11

Ожидаемые выходные данные: Ошибка: размер массива должен быть в диапазоне 0 <= n <= 10. Запустите программу еще раз!

Фактические выходные данные: Ошибка: размер массива должен быть в диапазоне 0 <= n <= 10. Запустите программу еще раз!

Исходный текст программы на языке C можно посмотреть, открыв файл [main.c](#)

Текст финальной программы на языке ассемблера, полученной после компиляции и с расширенными комментариями и оптимизацией можно посмотреть, открыв файл [main.S](#)

Текст программы на языке ассемблера, полученной после компиляции и с расширенными комментариями без оптимизации можно посмотреть, открыв файл [main_without_optimization.S](#)

Текст программы на ассемблере без комментариев, полученный сразу после компиляции программы на языке C, можно посмотреть, открыв файл [main_before.S](#) (Я не стал вставлять сюда тексты, чтобы не «захламлять» PDF файл)

Информация, подтверждающая выполнение задания в соответствии требованиями на **оценку 6**:

- Программа на ассемблере была скомпилирована при помощи соответствующих **аргументов командной строки** (gcc -O0 -Wall -masm=intel -S -fno-asynchronous-unwind-tables -fcf-protection=none main.c), а также за счёт **ручного** редактирования исходного текста ассемблерного кода программы;
- В реализованной программе использованы функции readArray, createArray и printArray с передачей данных через параметры;
- Добавлены исчерпывающие комментарии, поясняющие эквивалентное представление переменных в программе на C, а также описывающие передачу фактических параметров и перенос возвращаемого результата в функциях;
- Представлен набор тестов, на которых корректно отработала программа как на языке C, так и на ассемблере;
- Максимально использованы регистры процессора вместо локальных переменных (и не только), чтобы увеличить производительность. Подробнее:
 - Убраны команды por, которые буквально «ничего не делают»;
 - Операция «зануления» регистра оптимизирована при помощи xor;
 - Использование «нестираемого» 4-х байтового регистра r12d вместо локальной переменной -4[rbr] (для переменной i в коде на C)

- Использование «нестираемого» 4-х байтового регистра r13d вместо локальной переменной -100[rbp] (для переменной array_size в коде на C), однако для взятия указателя все-таки один раз -100[rbp] должна использоваться с командой lea;
- Удаление ненужных операций mov для перемещения значения из одного регистра на другой, которые используют регистр eax в качестве «посредника».