Отчёт по ИДЗ №1

Кормилицын Владимир Алексеевич, БПИ 226

Вариант 21

**Цель задания:**

Условие:

Сформировать отсортированный по возрастанию массив B из элементов массива A.

Разработать программу, которая вводит одномерный массив A, состоящий из N элементов (значение N вводится при выполнении программы), после чего формирует из элементов массива A новый массив B по правилам, указанным в варианте, и выводит его. Память под массивы может выделяться статически, на стеке, автоматически по выбору разработчика с учетом требований к оценке работы. При решении задачи необходимо использовать подпрограммы для реализации ввода, вывода и формирования нового массив. Допустимы (при необходимости) дополнительные подпрограммы. Максимальное количество элементов в массиве не должно превышать 10 (ограничение обуславливается вводом данных с клавиатуры). При этом необходимо обрабатывать некорректные значения как для нижней, так и для верхней границ массивов в зависимости от условия задачи.

**Отчёт о выполнении:**

Для выполнения задачи был разработан набор подпрограмм и макросов (macros):

1. Подпрограмма main в файле main.s, которая является точкой входа в программу.
2. Подпрограмма heap\_sort в файле heap\_sort.s для сортировки массива с использованием алгоритма пирамидальной сортировки (Heap sort).
3. Подпрограмма run\_auto\_tests для тестирования программы с использованием тестовых массивов
4. В библиотеке макросов stack\_tools.i реализованы макросы для работы со стеком:
   1. push
   2. pop
5. В библиотеке макросов console\_tools.i реализованы макросы для работы с вводом-выводом:
   1. read\_uint\_a0\_less\_eq\_imm
   2. read\_int\_a0\_with\_hint
   3. print\_int
   4. print\_int\_imm
   5. print\_int\_a0
   6. puts
   7. put\_char\_imm
   8. put\_newline
   9. exit\_imm
   10. exit
6. В библиотеке array\_tools.i реализованы макросы для работы с целочисленными массивами:
   1. input\_array
   2. print\_array
   3. read\_at
   4. write\_at
   5. array\_copy
   6. array\_equals

Для подключения библиотек макросов в единицах компиляции (файлах с расширением .s) используется директива .include

Для возможности вызова подпрограммы сортировки вне файла heap\_sort.s метка heap\_sort (адрес начала подпрограммы) помечена глобальной с помощью директивы .global

Для тестирования в подпрограмме run\_auto\_tests используются 5 заранее заданных массивов, которые после сортировки сравниваются с их заранее отсортированными аналогами. По ходу тестирования в выходной поток пишутся логи вида “Test i passed.” или “Test i not passed.”, где i – номер теста

**Исходный код программы:**

”main.s”:

Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

”stack \_tools.i”:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

”console \_tools.i”:

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

“array\_tools.i”:

Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

“auto\_tests.s”

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

“heap\_sort.s”:

Изображение выглядит как текст, снимок экрана, документ, меню

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, документ, Шрифт

Автоматически созданное описание

(также исходный код можно найти в файлах main.s, console\_tools.i, stack\_tools.i, array\_tools.i, auto\_tests.s и heap\_sort.s, находящихся в одном архиве с этим отчётом)

**Скриншоты работы программы:**

Изображение выглядит как текст, снимок экрана, документ, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, документ

Автоматически созданное описание