**Вариант 4, Задание 23**

Реализовать контейнер для хранения альтернатив и их параметров.

Обобщённый артефакт: объемные трехмерные геометрические фигуры.

Базовые альтернативы:

**· Шар** - целочисленный радиус.

**· Параллелепипед** - целочисленная длина первого ребра, целочисленная длина второго ребра, целочисленная длина третьего ребра.

**· Правильный тетраэдр** – длина ребра.

Общая для всех альтернатив переменная – плотность материала фигуры – действительное число.

Общая для всех альтернатив функция: вычисление площади поверхности – действительное число.

Обработка данных в контейнере: переместить в начало контейнера те элементы, для которых значение, полученное с использованием функции, общей для всех альтернатив, больше чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции. Остальные элементы сдвинуть к началу без изменения их порядка.

**Входные данные:**

Форматы входной команды:

· «./task -f <файл с входными данными> <файл для вывода фигур в контейнере> <файл для вывода отсортированных согласно заданию 23 фигур в контейнере>»

Например, ‘./task -f input.txt output.txt outputsorted.txt’.

· «./task -n <количество фигур для генерации> < файл для вывода фигур в контейнере > < файл для вывода отсортированных согласно заданию 23 фигур в контейнере >»

Например, ‘./task -n 50 output.txt outputsorted.txt’.

Формат описания фигур в файле входных данных:

**· Шар:**

1 <плотность материала> <целочисленный радиус>

Например, 1 10.0 5 - шар плотностью 10.0 радиуса 5.

**· Параллелепипед:**

2 <плотность материала> <длина первого ребра> <длина второго ребра> < длина третьего ребра>

Например, 2 15.3 3 4 5 - параллелепипед плотностью 15.3 с ребрами длиной 3, 4 и 5.

**· Правильный тетраэдр:**

3 <плотность материала> <длина ребра>

Например, 3 7.5 5 – правильный тетраэдр плотностью 7.5 и длиной ребра 5.

**Выходные данные:**

Вывод информации о контейнере:

Container contains <количество фигур в контейнере> фигур:

- **для Шара**:

Ball: density = <плотность материала фигуры>, radius = <радиус шара>, surface area = <площадь поверхности>

- **для Параллелепипеда**:

Parallelepiped: density = <плотность материала фигуры>, first edge length = <длина первого ребра>, second edge length = <длина второго ребра>, third edge length = <длина третьего ребра>, surface area = <площадь поверхности>

- **для Тетраэдра**:

Tetrahedron: density = <плотность материала фигуры>, edge length = <длина ребра>, surface area = <площадь поверхности>

Вывод среднего арифметического площадей поверхностей:

Average surface area = <среднее арифметическое площадей поверхностей фигур в контейнере>

**Метрики**

Состав: 5 файлов реализации: main.c; output.c; random\_input.c; file\_input.c; area.asm.

1 заголовочный файл: data.h

Размер файлов: **12 КБ**

Время исполнения:

**20 фигур**

Аргументы:

./task -n 20 tests/randomtest/output.txt tests/randomtest/outputsorted.txt

Время: **3 млс.**

**100 фигур**

Аргументы:

./task -n 100 tests/randomtest/output.txt tests/randomtest/outputsorted.txt

Время: **3 млс.**

**500 фигур**

Команда:

./task -n 500 tests/randomtest/output.txt tests/randomtest/outputsorted.txt

Время: **4 млс.**

**2000 фигур**

Аргументы:

./task -n 2000 tests/randomtest/output.txt tests/randomtest/outputsorted.txt

Время: **8 млс.**

**10000 фигур**

Аргументы:

./task -n 10000 tests/randomtest/output.txt tests/randomtest/outputsorted.txt

Время: **31 млс.**

Сравнение:

Программа с использованием ассемблера выполняется в разы быстрее, т.к. не происходит издержек из-за преобразования языка высокого уровня в машинный код. Исполняемые файлы написанные на ассемблере весят меньше.