Архитектура вычислительных систем.

Задание 1. Вариант 12. Функция 19.

Выполнила: Словягина Анна Михайловна

Студентка 2-го курса программной инженерии

группа БПИ207

1 Описание задания

* Обобщенный артефакт, используемый в задании:
  + животные.
* Базовые альтернативы и их уникальные параметры:
  + рыбы:

место проживания - перечислимый тип (река, озеро, океан, море)

* + птицы:

отношение к перелету - булевская величина (перелетные, не перелетные)

* + звери:

тип зверя - перечислимый тип (травоядные, хищники, насекомоядные)

* Общие для всех альтернатив переменные:
  + Название - строка символов;
  + Вес в граммах - целое число.
* Общие для всех альтернатив функции:
  + Частное от деления суммы кодов названия животного на вес - действительное число.
* Функция - удаление из контейнера тех элементов, для которых значение, полученное с использованием функции, общей для всех альтернатив, меньше, чем среднее арифметическое для всех элементов контейнера, полученное с использованием этой же функции.

2 Структурная схема изучаемой архитектуры ВС

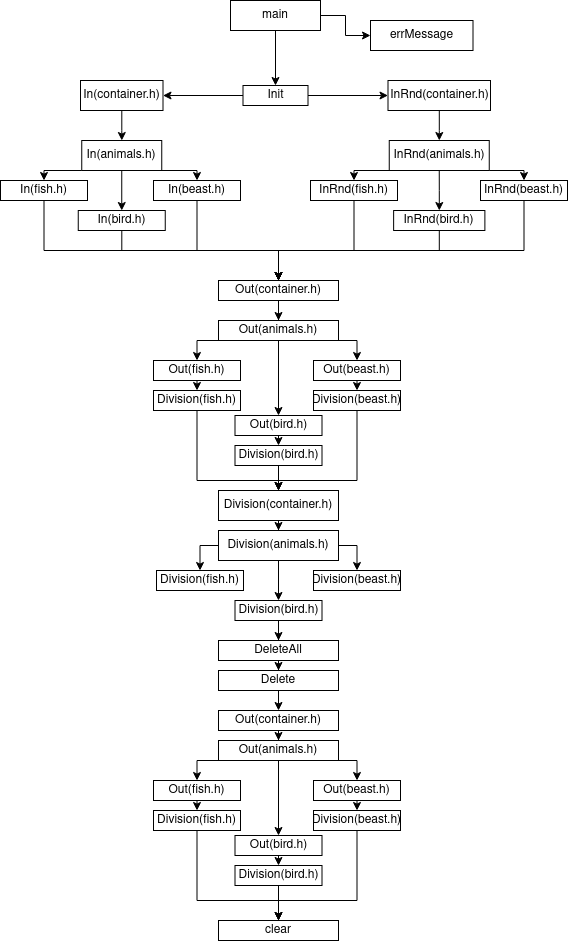
Таблица типов

|  |  |
| --- | --- |
| int | 4 |
| double | 8 |
| char | 1 |
| bool | 1 |
| Struct container  enum max\_len  int: len  animals: cont[max\_len] | 280008  4[0]  4[4]  28\*10000[8] = 280000[8] |
| Struct animals  enum key  k: key  union  fish: f  bird: b  beast: be | 28  4[0]  24[4]  24[4]  24[4] |
| Struct fish  name: char[15]  weigh: int  enum keyF  k: keyF | 24  15\*1[0] = 15[0]  4[15]  4[19] |
| Struct bird  name: char[15]  weigh: int  migratory: bool | 24  15\*1[0] = 15[0]  4[15]  1[19] |
| Struct beast  name: char[15]  weigh: int  enum type  k: type | 24  15\*1[0] = 15[0]  4[15]  4[19] |

Память программы

|  |  |
| --- | --- |
| main(int argc, char\* argv[])  argc: int  argv: char\*  ifst: ifstream  container: struct container  size: auto  ofst1: ifstream  ofst2: ifstream | 4[0]  1[4]  520[5]  280008[525]  4[280533]  512[280537]  512[281049] |
| void DeleteAll(container &c)  c: struct container  count: int | 280008[0]  4[280008] |
| double Division(container &c)  c: struct container  sum: double  count: int | 280008[0]  8[280008]  4[280016] |
| void InRnd(bird &b)  b: struct bird  name\_length: int  migratory: int | 24[0]  4[24]  4[28] |

3 Стек



4 Основные характеристики программы

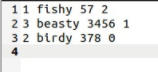
Число интерфейсных модулей: 5

Число модулей реализации: 6

Общий размер исходных текстов: 589

Формат ввода:

В каждой строке файла должно содержаться 4 переменные, разделенные пробелом:



Первая переменная отвечает за вид животного (1 – рыба, 2 – птица, 3 – зверь), вторая переменная – за название животного, третья переменная – за его вес в граммах, четвертая переменная – за уникальные параметры.

Уникальные параметры для рыбы – это место обитания:

1 – река

2 – озеро

3 – океан

4 – море

Уникальный параметр для птицы – переменная, которая отвечает за то, является ли птица перелетной

0 – нет

1 – да

Уникальные параметры для зверя – это тип животного

1 – травоядный

2 – хищный

3 – насекомоядный

Расположение входных и выходных данных

Входные данные лежат в:

HomeWork\_1/cmake-build-debug/tests

Выходные для первого задания:

HomeWork\_1/cmake-build-debug/results1

Выходные для второго задания:

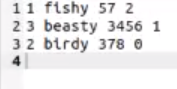
HomeWork\_1/cmake-build-debug/results2

5 Время выполнения программы для различных тестовых наборов данных

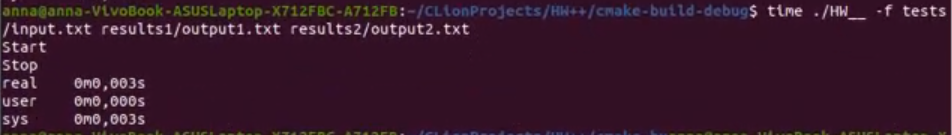
Ввод из файла:

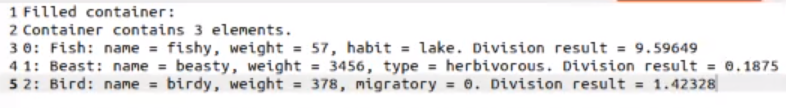
1 тест:

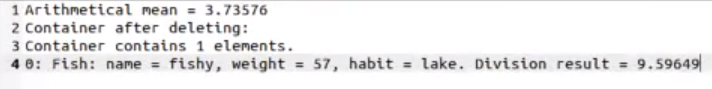
Ввод:



Запуск программы:



Результат:

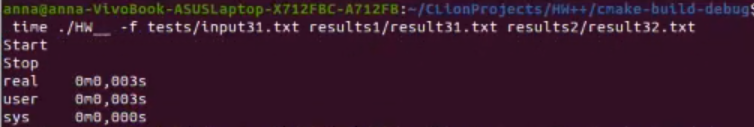


2 тест:

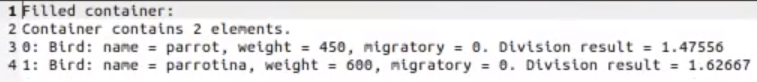
Ввод:

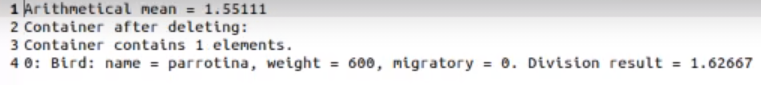


Запуск программы:



Результат:

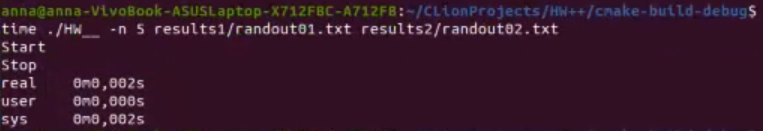




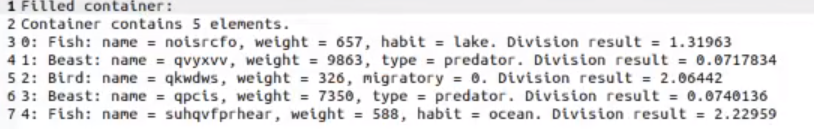
Рандомный ввод:

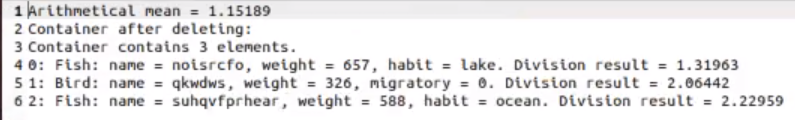
3 тест:

Запуск программы:



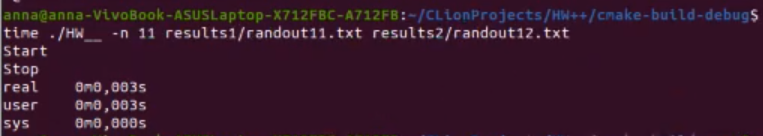
Результат:



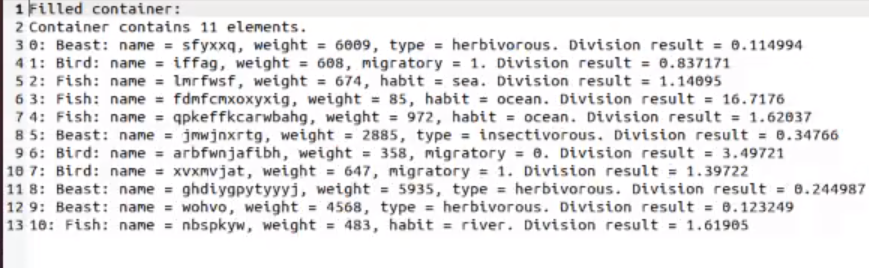


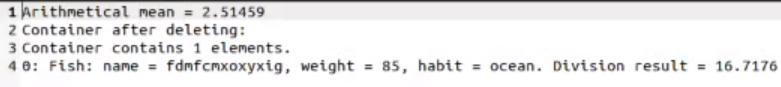
4 тест:

Запуск программы:



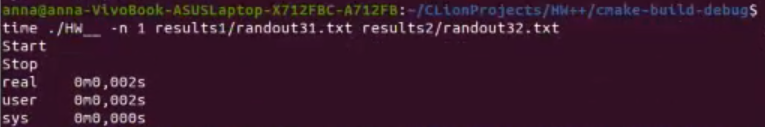
Результат:





5 тест:

Запуск программы:



Результат:

