

Вариант: **222**

Задание: **12**

Функция: **16**

### **Описание задачи:**

Необходимо реализовать программу, в которой обобщенным артефактом является животное, а базовыми альтернативами: рыбы (поле – место проживания – перечислимый тип: река, озеро, море, пруд, океан), птицы (параметр – отношение к перелёту – перелётные или остающиеся на зимовку), звери (поле – тип питания – хищники, травоядные, насекомоядные, всеядные).

Общим для все альтернатив являются: имя животного/его название – строка символов, вес животного в граммах – целое число.

Общей для всех альтернатив функцией является нахождение частного от деления суммы кодов незашифрованной строки на вес, являющиеся действительным числом.

Обработка данных является их упорядочивание по убыванию, с использованием сортировки методом деления по пополам (*Binary Search*).

### **Основные характеристики программы:**

Число интерфейсных модулей: **5**

Число модулей реализация: **6**

Общий размер исходных текстов: **~18,12 КБ**

Размер исполняемого файла: **103 КБ**

Время работы на тестах:

TEST 1: **0.003 с**

TEST 2: **0.012 с**

TEST 3: **0.001 с**

(\*программа завершилась преждевременно из-за некорректных входных данных)

TEST 4: **0.005 с**

TEST 5: **0.001 с**

(\*программа завершилась преждевременно из-за некорректных входных данных)

TEST 6: **0.013 с**

TEST 7: **0.001 с**

(\*программа завершилась преждевременно из-за некорректных входных данных)

TEST 8: **0.009 с**

TYPE TABLE	
INT	4
DOUBLE	8
CHAR*	8
enum AnimalType: BEAST, BIRD, FISH	4[0]
enum Nutrition: CARNIVORES, HERBIVORES, INSECTIVORES, OMNIVORES	4[0]
enum Habitat: RIVER, SEA, LAKE, OCEAN, POND	4[0]
struct Beast: nutrition: Nutrition	4 4[0]
struct Bird: is_migratory: int	4 4[0]
struct Fish: habitat: Habitat	4 4[0]
struct Animal: name: char* weight: int key: AnimalType union: beast : Beast bird: Bird fish: Fish	40 8[0] 12[8] 16[12]  4[12] 4[12] 4[12]
struct Container: int: size Animal*: all_animals	44 4[] 40[4]

## PROGRAM MEMORY

<b>main (int argc, char* argv[]):</b> argv: int argc: char** size: int container: Container input: ifstream general_output: ofstream sorted_output: ofstream	<b>1604</b> 4[0] 8[4] 4[12] 44[16] 520[60] 512[580] 512[1092]
<b>Quotient (Animal &amp;animal):</b> sum: double i: int	<b>12</b> 8[0] 4[8]
<b>InputRandom(Container &amp;container):</b> letters: char* i: int	<b>12</b> 8[0] 4[8]
<b>InputFromFile (Container &amp;container, ifstream &amp;input):</b> i: int animal: Animal	<b>44</b> 4[0] 40[4]
<b>SortContainerDataWithInsertionSort (Container &amp;container):</b> i: int loc: int j: int n: int all_animals: Animal* selected: Animal	<b>96</b> 4[0] 4[4] 4[8] 4[12] 40[16] 40[56]

### Дополнительная информация:

Данных, демонстрирующих сравнение с характеристиками ранее разработанных программ нет, так как это первый проект.

Программа была выполнена на C++ в стиле C.

Подробная информация о корректном использовании находится в файле README.txt

### Ниже представлен ПРИБЛИЗИТЕЛЬНЫЙ стек вызовов:

Звездочками помечены методы, которые вызываются более одного раза в зависимости от параметров.

