**Бутченко Даниил**

**БПИ206**

**Отчет**

Номер варианта задания: 205

Номер условия задачи: 9

Номер дополнительной функции: 15

Условия задачи:

Обобщенный артефакт, используемый в задании: тексты, состоящие из цифр и латинских букв, зашифрованные различными способами.

Базовые альтернативы:

1. Шифрование заменой символов.
2. Шифрование циклическим сдвигом кода каждого символа на n.
3. Шифрование заменой символов на числа.

Общие для всех альтернатив переменная: Открытый текст – строка символов.

Общие для всех альтернатив функция: Частное от деления суммы кодов незашифрованной строки на число символов в этой строке

Дополнительная функция:

Упорядочить элементы контейнера по убыванию используя сортировку с помощью «дерева» (Heap Sort). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Характеристики программы:

Число интерфейсный модулей: 0

Число модулей реализации: 7

Общий размер исходных текстов: 21526 байт

Размер исполняемого кода: 0 байт

Время выполнения программы для тестовых наборов данных:

Test 1: *0.002 сек. (Ввод из файла)*

Test 2: *0.001 сек. (Ввод из файла)*

Test 3: *0.003 сек. (Ввод из файла)*

Test 4: *0.006 сек. (Ввод из файла)*

Test 5: *0.001 сек. (Ввод из файла)*

Test 6: *0.04 сек. (Случайная генерация)*

Test 7: *4.909 сек. (Случайная генерация)*

Сравнительный анализ по полученным характеристикам с

реализациями, сделанными в других заданиях:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| Число интерфейсных модулей | 5 | 5 | 0 |
| Число модулей реализации | 6 | 6 | 7 |
| Общий размер исходных текстов | 21802 | 22161 | 21526 |
| Размер исполняемого кода | 111104 | 114176 | 0 |
| Test 1 | *0.02 сек.* | *0.01 сек.* | *0.002 сек.* |
| Test 2 | *0.022 сек.* | *0.003 сек.* | *0.001 сек.* |
| Test 3 | *0.004 сек.* | *0.004 сек.* | *0.003 сек.* |
| Test 4 | *0.085 сек.* | *0.009 сек.* | *0.006 сек.* |
| Test 5 | *0.025 сек.* | *0.003 сек.* | *0.001 сек.* |
| Test 6 | *0.021 сек.* | *0.007 сек.* | *0.04 сек.* |
| Test 7 | *0.499 сек.* | *0.531 сек.* | *2.809 сек.* |

Вывод: на малом количестве данных динамическая типизация работает быстрее, однако для большого количества данных она серьезно уступает статической типизации, как для ооп, так и для процедурного подхода.

Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как стол

Автоматически созданное описание

Stack вызовов:

