**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: 5

Число модулей реализации: 6

Общий размер исходных текстов: 21802 байт

Размер исполняемого кода: 111104 байт

Время выполнения программы для тестовых наборов данных:

Test 1: *0.02 сек*

Test 2: *0.022 сек*

Test 3: *0.004 сек*

Test 4: *0.085 сек*

Test 5: *0.025 сек*

Test 6: *0.021 сек.*

|  |  |
| --- | --- |
| Типы данных | |
| INT | 4 |
| SHORT | 2 |
| CHAR | 1 |
| DOUBLE | 8 |
| struct CyclicShiftEncryption {  char encryptedString[256];  short shift;  } | 258  256\*1 = 256[0]  2[256] |
| struct NumbesEncryption {  int encryptedString[256];  PairOfCharInt \*symbols;  } | 1029  256\*4 = 1024[0]  5[1024] |
| struct PairOfCharInt {  char first;  int second;  } | 5  1[0]  4[1] |
| struct SymbolsEncryption {  char encryptedString[256];  PairOfCharChar \*symbols;  } | 258  256\*1 = 256[0]  2[256] |
| struct PairOfCharChar {  char first;  char second;  } | 2  1[0]  1[1] |
| struct Encryption {  key k;  union {  CyclicShiftEncryption c;  NumbersEncryption n;  SymbolsEncryption s;  }  Char sourceString[256];  Int size;  } | 1293  4[0]  258[4]  1029[4]  258[4]  1\*256 = 256[1033]  4[1289] |
| enum key {  Cyclic,  Numbers,  Symbols  } | 4 |
| struct container{  int len;  Encryption \*cont[10000];  } | 40004  4[0]  4\*10000 = 40000[4] |

|  |  |
| --- | --- |
| Память программы | |
| main (int argc, char\* argv[]) {  int argc;  char\*\* argv;  container c;  ifstream isft;  ofstream ofst1;  ofstream ofst2;  int size;  } | 41560  4[0]  4[4]  40004[8]  520[40012]  512[40532]  512[41044]  4[41556] |
| QuotientOfDivision(Encryption &e){  Encryption &e;  double sum;  int i;  } | 16  4[0]  8[4]  4[12] |
| HeapSort(container &c) {  int i;  int j;  Heapify  { | 4[0]  4[4]  40020[8] |
| Heapify(Encryption \*arr[10000],int n, int i) {  int n;  int i;  Encryption \*arr[10000];  Int smallest;  Int left;  Int right; | 40020  4[0]  4[4]  4\*10000 = 40000[8]  4[40008]  4[40012]  4[40016] |

Stack вызовов:

