

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Московский государственный технический
университет имени Н. Э. Баумана (национальный исследовательский
университет)

Факультет «Робототехника и комплексная автоматизация»
Кафедра «Системы автоматизированного проектирования»

Отчет по лабораторной работе

По курсу «Объектно-ориентированное программирование»

Выполнил:

Студент Петраков С.А.
Группа РК6-26Б

Проверил:

Дата _____
Подпись _____

Москва, 2020 г.

ООП Вариант 16А

Задание:

Разработать объектно-ориентированную программу подсчета отсутствующих строчных букв для любого набора латинских букв, который задан аргументом командной строки её вызова. Программная реализация вычислений должна быть основана на разработке класса подмножества латинских букв, состав которого кодирует двоичные разряды приватного массив из пары целых чисел без знака. Конструктор этого класса должен обеспечивать его инициализацию по любой заданной и пустой строке или алфавиту строчных букв (a-z). Кроме того, в нём должна быть предусмотрена публичная перегрузка оператора (), чтобы эффективно вычислять расстояние Хэмминга для бинарных кодов алфавита строчных букв и заданного набора с маскировкой его строчных букв компонентной перегрузкой оператора &. При этом в операторе () должен быть реализован быстрый подсчет 1 разрядов, а его вызов должен обеспечивать пустой набор Н. Результат вычислений должен отображаться строкой стандартного вывода. Печать операндов расчетной формулы должен обеспечивать оператор преобразования типа латинского набора в строку его символов (char *).

Алгоритм:

Для решения данной программы я создал класс, в котором хранится массив из 2 элементов, которые хранят двоичные числа для кодировки 2 символьных строк, состоящих только из строчных и только из заглавных букв, также хранятся компонентные методы перегрузки операторов “(char *)”, “&” и “()”. Чтобы эффективно рассчитать количество отсутствующих букв в строке, я нахожу расстояние Хэмминга для бинарных кодов алфавита строчных букв.

Входные данные:

1 символьная строка, состоящая из латинских букв

Выходные данные:

Количество отсутствующих строчных букв в строке

Текст программы:

```
#include <stdlib.h>
#include <iostream>
using namespace std;

class Alpha
{
private:
    unsigned int bin[2];
    int Calc(unsigned int b); //a quick count of bits
public:
    Alpha();
    Alpha(char*);
    operator char* (); // Overload (char*)
    int operator () (Alpha&, Alpha&); // Overload ()
    Alpha operator&(unsigned int*); // Overload &
```

```

};
Alpha::Alpha() // Default constructor
{
    //Full alphabet
    bin[0] = 67108863;
    bin[1] = 0;
}
Alpha::Alpha(char* str) // Constructor with string
{
    bin[0] = 0;
    bin[1] = 0;
    while (*str)
    {
        if (*str >= 'a')
            bin[0] |= (1 << (*str - 'a'));
        else
            bin[1] |= (1 << (*str - 'A'));
        str++;
    }
}
Alpha::operator char* ()
{
    static char s[64];
    int j = 0;
    for (int n = 0; n < 2; n++)
    {
        unsigned int b = bin[n];
        int i = 0;
        while (b > 0)
        {
            if (b & 1)
                s[j++] = 'a' + i - n * 32;
            i++;
            b >>= 1;
        }
        s[j] = 0;
        return (s);
    }
}
int Alpha::Calc(unsigned int b)
{
    int i = 0;
    while (b != 0)
    {
        b = b & (b - 1);
        i++;
    }
    return (i);
}

int Alpha::operator () (Alpha& a, Alpha& b)
{
    return Calc(a.bin[0] ^ b.bin[0]) + Calc(a.bin[1] ^ b.bin[1]);
}
Alpha Alpha::operator&(unsigned int* a)
{
    Alpha result;
    result.bin[0] = this->bin[0] & a[0];
    result.bin[1] = this->bin[1] & a[1];
    return result;
}

int main(int argc, char* argv[])

```

```

{
    // Verifying arguments
    if (argc != 2)
    {
        cout << "Not enough arguments." << endl;
        return -1;
    }

    //Prog Start
    Alpha first(argv[1]);
    Alpha second;
    char s[] = "";
    Alpha H(s);

    //Hemming count
    Alpha temp;
    unsigned int mask[2] = { 67108863,0 };
    temp = first & mask;

    cout << "H(" << (char*)temp << ", ";
    cout << (char*)second << ")= " << H(temp, second) << "." << endl;
    return (0);
}

```

Тесты:

Номер	Вход	Выход
1	abc	23
2		Not enough arguments.

Список использованной литературы:

- Волосатова Т.М., Родионов С.В. Лекции по курсу «Объектно-ориентированное программирование»
- bigor.bmstu.ru