

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация ЭВМ и систем»
Тема: Представление и обработка символьной информации с
использованием строковых команд.

Студентка гр. 9383

Орлов Д.С

Преподаватель

Ефремов
М.А.

Санкт-Петербург

2020

Цель работы.

Изучить представление и обработку символьной информации — строки символов на языке Ассемблер.

Написать программу, обрабатывающую строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

Текст задания.

Разработать программу обработки символьной информации, реализующую функции:

- инициализация (вывод титульной таблички с указанием вида преобразования и автора программы) - на ЯВУ;
- ввода строки символов, длиной не более N_{\max} (≤ 80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает N_{\max} , остальные символы следует игнорировать;
- выполнение заданного в таблице 5 преобразования исходной строки с записью результата в выходную строку - на Ассемблере;
- вывода результирующей строки символов на экран и ее запись в файл - на ЯВУ.

Ассемблерную часть программы включить в программу на ЯВУ по принципу встраивания (in-line).

Исходные данные.

Вариант 14

Исключение латинских букв и цифр, введенных во входной строке при формировании выходной строки.

Ход работы.

В ходе работы была разработана программа на языке C++ и Ассемблер, которая исключает все латинские буквы и цифры из входной строки.

Исключение происходит следующим образом: все символы, имеющие код 1 - 47, 58 - 64, 91 - 96 и больше 123, записываются в выходную строку. Остальные символы игнорируются.

В файле *lb4.cpp* находится три функции:

char MString (char* str_1)* — функция, которая принимает на вход исходную строку, преобразовывает как указано в задании и возвращает новую строку.

int main() - основная функция, в которой выделяется память под исходную строку, после чего происходит считывание. В ней же происходит вызов функции и вывод выходной строки в консоль.

Исходный код и листинг программы представлены в приложении А.

Примеры работы программы.

Таблица 2 — Примеры работы программы.

Входная строка	Выходная строка
1a263в4г	абвг
Рйw3ц2sждf1	йцж
1!й»tre213vШ	!й»Ш

Выводы.

Было изучено представление и обработка символьной информации — строки символов на языке Ассемблер.

Была написана программа, обрабатывающая строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb4.cpp

```
#include <iostream>

int main()
{
    char* str = new char[80];
    std::cout << "Входная строка:\n";
    std::cin.getline(str, 80);
    char* str_2 = new char[80];

    asm(
        "mov rsi, %0\n"
        "mov rdi, %1\n"

        "loop:\n"
            "mov ah, [rdi]\n"
            "cmp ah, 0\n"
            "je exit\n"

            "cmp ah, 0x30\n"
            "jge is_digit\n"

            "jmp write\n"

        "is_digit:\n"
            "cmp ah, 0x39\n"
            "jle miss\n"

            "cmp ah, 0x41\n"
            "jge is_latin_big\n"

            "jmp write\n"

        "is_latin_big:\n"
            "cmp ah, 0x5A\n"
            "jle miss\n"

            "cmp ah, 0x61\n"
            "jge is_latin_small\n"

            "jmp write\n"

        "is_latin_small:\n"
            "cmp ah, 0x7A\n"
            "jle miss\n"

            "jmp write\n"

        "miss:\n"
            "inc rdi\n"
            "jmp loop\n"
```

```

        "write:\n"
            "mov [rsi], ah\n"
            "inc rsi\n"
            "inc rdi\n"
            "jmp loop\n"

    "exit:\n"

    : "=m"(str_2)
    : "m"(str)
    );

std::cout << "Орлов Даниил, группа 9383, вариант 14\n"
    << " Исключение латинских букв и цифр, введенных "
    "во входной строке при формировании выходной строки" << std::endl;

std::cout << "Результат: " << str_2 << std::endl;

delete[] str;
delete[] str_2;
return 0;
}

```