

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «Организация ЭВМ и систем»
ТЕМА: ПРЕДСТАВЛЕНИЕ И ОБРАБОТКА СИМВОЛЬНОЙ ИНФОРМАЦИИ С
ИСПОЛЬЗОВАНИЕМ СТРОКОВЫХ КОМАНД.

Студентка гр. 0382

Чегодаева Е.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить представление и обработку символьной информации с использованием строковых команд.

Задание.

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).

Вариант 3: Формирование выходной строки только из русских и латинских букв входной строки.

Выполнение работы.

Работа выполнена на языке C++ с использованием блока `__asm`, который реализует все необходимые действия над строкой на Ассемблере по принципу встраивания.

Для возможности взаимодействия с русскими буквами подключена функция `setlocale()`. Далее реализован вывод титульной таблички с указанием вида преобразования и автора программы. Посредством `fgets()` программа получает строку, длина которой не превышает 80 символов (81, включая символ окончания строки).

Операции блока `__asm`:

Выполнено присваивание, посредством оператора `offset`, регистрам `esi` и `edi` смещения на исходную строку и на строку результата соответственно.

Check — блок проверки поступающего символа:

Посредством команды `lodsb` в регистр `AL` записывается символ из `esi`.

Первостепенно осуществляется проверка на символ окончания строки при помощи `str` и условного оператора `je(=)`, в случае встречи — реализован переход к блоку `END`.

Далее реализованы три “типа” поступающих символов, которые должны быть в результирующей строке: “A-z”, “A-я”, “Ё,ё”. Русские символы — Если встретился символ, код которого по таблице `ASCII` больше либо равен коду первой буквы алфавита (заглавной) и при этом меньше либо равен последней (строчной), то вызывается переход к блоку `Res`, иначе данный символ игнорируется и вновь осуществляется переход в `Check`. Отдельно рассмотрены символы “Ё” и “ё”, так как они в таблице `ASCII` отделены от основного алфавита. Аналогично реализовано чтение латинских букв из строки.

Res — блок записи символа:

Посредством команды `stosb` сохраняется значение регистра `AL` (т.е. текущего символа) в `edi`.

Далее реализован переход к `Check`.

END — блок окончания обработки строки.

После окончания работы блока `__asm` — строка, которая хранится в `edi` будет выведена на экран и записана в файл “`Resalt.txt`”.

Исходный код программы см. в приложении А.

Тестирование.

Результаты тестирования представлены в таблице 1.

Таблица 1 – результаты тестирования.

№	Входные данные	Выходные данные	Комментарии
1	HelloПривет123	HelloПривет	Верно
2	I очень like КОФЕ!!!	ЮченьlikeКОФЕ	Верно
3	Ёё55Ss55Лл..	ЁёSsЛл	Верно

Выводы.

Были изучены представление и обработка символьной информации с использованием строковых команд. Реализована программа, которая формирует и выводит строку, состоящую только из русских и латинских букв введённой пользователем строки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММ

Название файла: lb4.cpp

```
#include <iostream>
#include <fstream>
#include <stdio.h>

char input[81];
char output[81];

int main() {
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");
    std::cout << "0382 Chegodaeva Elizaveta.\nVar№3: Only Russian and Latin letters\n";
    fgets(input, 81, stdin);
    input[strlen(input)] = '\0';
    std::ofstream file("resalt.txt");
    __asm {
        mov esi, offset input
        mov edi, offset output

        Check:
            lodsb
            cmp al, '\0'
            je END

            cmp al, 'Ё'
            je Rec
            cmp al, 'ё'
            je Rec

            cmp al, 'А'
            jl Check
            cmp al, 'я'
            jle Rec

            cmp al, 'А'
            jl Check
            cmp al, 'з'
            jle Rec

        Rec:
            stosb
            jmp Check

        END:

    };
    std::cout << output;
    file << output;
    return 0;
}
```