МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРИТЕТ

имени М.В. ЛОМОНОСОВА

ФАКУЛЬТЕТЕ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

ОТЧЕТ ПО ЗАДАНИЮ № 2

**«Сборка многомодульной программы.**

**Обработка символьных данных»**

**Вариант 5 / 6 / 1**

Выполнил:

студент 141 группы

Зеленчук Георгий Владимирович

Преподаватель:

Кузьменкова Евгения Анатольевна

Москва

2021

Оглавление

[Постановка задачи 3](#_Toc70776652)

[Структура программы и спецификации функций 3](#_Toc70776653)

[Сборка программы 4](#_Toc70776654)

[Отладка программы, тестирование функций 4](#_Toc70776655)

[Анализ допущенных ошибок 4](#_Toc70776656)

[Литература 5](#_Toc70776657)

# Постановка задачи

Дан непустой текст (последовательность символов, отличных от пробела), содержащий не более 100 элементов, с точкой в качестве признака конца текста (точка в текст не входит).

Требуется:

* Ввести с клавиатуры данный текст и записать его в память.
* Определить, обладает ли этот текст следующем свойством: текст содержит равное количество заглавных и строчных латинских букв.
* Преобразовать текст по правилу 1[заменить каждую заглавную латинскую букву на заглавную букву, симметричную ей в алфавите (A ↔ Z, B ↔ Y, …)], если он обладает заданным свойством, и по правилу 2[перенести в начало текста все входящие в него цифры с сохранением порядка их следования] в противном случае.
* Вывести: исходный текст, номер примененного правила, преобразованный текст.

Задача выполняется в виде многомодульной программы. Проверка выполнения данного свойства выполняется функцией, реализованного на языке программирования Си. Преобразование текста по правилам 1 и 2 выполняется функциями, описанными на языке ассемблера NASM.

# Структура программы и спецификации функций

Файл task.c содержит в себе основную часть программы на языке Си, в нём использованы следующие функции:

* int main() – Основная функция. Читает строку text, проверяет выполнение заданного свойства, вызывает соответствующую функцию преобразования строки, после чего выводит преобразованную строку.
* void rule1(char\*) – Функция, преобразующая заданную строку по правилу: заменить каждую заглавную латинскую букву на заглавную букву, симметричную ей в алфавите (A ↔ Z, B ↔ Y, …).
* void rule2(char\*) - Функция, преобразующая заданную строку по правилу: перенести в начало текста все входящие в него цифры с сохранением порядка их следования.

Файл func.h – заголовочный файл, описывающий функции, вызываемые для преобразования исходной строки.

Файлы rule1.asm и rule2.asm содержат в себе описание одноимённых функций на языке NASM, выполняющих преобразование строки по заданным правилам.

task.c

include “func.h”

. . .

func.h

void rule1(char\*);

void rule2(char\*);

rule1.asm rule2.asm

section .txt

global rule2

rule2:

...

section .txt

global rule1

rule1:

...

# Сборка программы

* nasm -f elf32 -o rule1.o rule1.asm – создания объектного файла функции для преобразование строки по первому правилу
* nasm -f elf32 -o rule2.o rule2.asm – создания объектного файла функции для преобразование строки по второму правилу
* gcc -Wall -m32 -o res task.c rule1.o rule2.o – компиляция всех модулей программы

# Отладка программы, тестирование функций

Тесты функций для преобразования текста по правилу 1:

|  |  |  |
| --- | --- | --- |
| Номер теста: | Входные данные: | Выходные данные: |
| 1 | . | \*пустой текст\* |
| 2 | 0Aa+Zz0. | 0Za+Az0 |
| 3 | Aa1Bb2c. | Aa1Bb2c |

Тесты функций для преобразования текста по правилу 2:

|  |  |  |
| --- | --- | --- |
| Номер теста: | Входные данные: | Выходные данные: |
| 1 | . | \*пустой текст\* |
| 2 | 0Aa+Zz0. | 0Aa+Zz0 |
| 3 | Aa1Bb2c. | 12AaBbc |

Тесты всей программы:

|  |  |  |
| --- | --- | --- |
| Номер теста: | Входные данные: | Выходные данные: |
| 1 | . | \*пустой текст\* |
| 2 | 0Aa+Zz0. | 0Za+Az0 |
| 3 | Aa1Bb2c. | 12AaBbc |
| 4 | 1AAA2aaa3. | 1ZZZ2aaa3 |
| 5 | A0. | 0A |

# Анализ допущенных ошибок

В функции rule2 были использованы методы работы с целочисленным массивом вместо массива символов. Для исправления потребовалось изменить размерность использованных регистров и способ изменения адреса массива.

# Литература

[1] Е.А. Кузьменкова, В.С. Махнычев, В.А. Падарян. Семинары по курсу «Архитектура ЭВМ и язык ассемблера »: учебно-методическое пособие.

[2] Руденко Т.В. Сборник задач и упражнений по языку Си.