Белорусский государственный университет

Факультет прикладной математики и информатики

Кафедра информационных систем управления

Поздняков Евгений Евгеньевич

**Отчет**

**по учебной практике**

студента 1 курса 8 группы

**Преподаватель**

*Карпович Наталья Александровна*

старший преподаватель

кафедры технологий программирования

Минск, 2015

Содержание

[1.Постановка задачи 3](#_Toc412579225)

[2. Алгоритм решения задачи п. 1.1 4](#_Toc412579226)

[3. Алгоритм решения задачи п. 1.2 4](#_Toc412579227)

[4. Реализация функций 5](#_Toc412579228)

[5.Примеры работы программы: 6](#_Toc412579229)

# 1.**Постановка задачи**

* 1. Задан текстовый файл Input.txt, состоящий из слов. Разделителями между словами является некоторое множество знаков препинания. Найти в каждой строке слова, состоящие из заглавных букв, удалить и дописать в начало строки. Результат записать в новый файл Output.txt. Упорядочить слова в полученных строках по убыванию длин слов, результат сортировки записать в файл Out\_sort.txt.
  2. Реализовать отдельные функции программы из п.1 с использованием цепочечных команд.

# 2. **Алгоритм решения задачи п. 1.1**

* 1. Выполним открытие файла «Input.txt» и проверку на его существование в директории с программой.
  2. Выполним чтение файла в строку
  3. Разобьем строку на лексемы
  4. Выполним проверку каждого слова на наличие заглавной буквы
  5. Создадим новую строку размером со считанную из файла
  6. Если слово содержит заглавную букву, заносим его в строку 1
  7. Если слово не содержит заглавную букву, заносим его в строку 2
  8. Шаги 2.6 и 2.7 будем выполнять пока в исходной строке не останется лексем (будет достигнут конец строки)
  9. Затем произведем сцепление строки 1(со словами с заглавными буквами) и строки 2(с остальными словами) в одну строку
  10. Полученную на шаге 2.9 строку занесем в файл Output.txt
  11. После выполнения операции на экране появится сообщение: "Uppercase sort complite, see Output.txt"
  12. Для выполнения сортировки создадим динамический массив строк (каждая строка одно слово)
  13. С помощью функции QuickSort выполним сортировку.
  14. Отсортированный массив запишем в файл Out\_sort.txt
  15. После выполнения операции на экране появится сообщение: "QuickSort complite, see Output\_sort.txt"

# 3. **Алгоритм решения задачи п. 1.2**

* 1. Реализуем функцию strlen, принимающую в качестве параметра массив символов (строку). Возвращаемое значение будет числом (количество символов)
  2. Заносим адрес строки в регистр edi
  3. Будем выполнять поиск символа конца строки с помощью команды repne scasb
  4. Количество символов будет содержаться в регисте eax
  5. Переносим данные из регистра eax в переменную
  6. Делаем возврат значения количества символов в строке

# **4. Реализация функций**

int strlen(char\* s)

{

int x;

\_asm

{

xor ecx, ecx

push edi

xor eax, eax

cld

dec ecx

mov edi, s

repne scasb

not ecx

pop edi

lea eax, [ecx - 1]

mov x, eax

}

return x;  
}

void strcat(char \*str1, char \*str2)

{

\_asm

{

mov edi, str1

push edi; //конец str1

mov edi, str2

mov esi, edi

not ecx//длина str2

pop edi

rep movsb; //Копированиt строк

xor al, al

stosb

mov eax, str1;

ret

xor ecx, ecx

mov al, cl

dec ecx

repne scasb

dec edi

}

}

# **5.Примеры работы программы:**

Input.txt

Mom Vava hghghghghg tytyty Dddjjd tititi gkkgkg kk kj

Output.txt:

Mom Vava Dddjjd hghghghghg tytyty tititi gkkgkg kk kj

Output\_sort.txt: kj kk Mom Vava gkkgkg tititi Dddjjd tytyty hghghghghg

Input.txt

Mom, Vava. hghghghghg! Hytyty Dddjjd tititi gkkgkg ???? Kk kj KKKK///

Output.txt:

Mom Vava Hytyty Dddjjd Kk KKKK hghghghghg tititi gkkgkg kj

Output\_sort.txt: kj Kk Mom KKKK Vava gkkgkg Dddjjd tititi Hytyty hghghghghg

