Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №4

Работа со строками

Выполнил: Студент: гр. 053502

Юрьев Вадим Андреевич

Руководитель: ст. преподаватель

Шиманский В.В.

Минск 2021

СОДЕРЖАНИЕ

1. Введение
2. Постановка задачи
3. Программная реализация
4. Выводы

Литература

Приложение

1. Введение

Целью данной работы является изучить следующий материал:

1. Представление строки в языке Ассемблер.
2. Команды movsb, movsw, stosb, stows, lodsb, lodsw.
3. Назначение флага направления, команды CLD и STD.
4. Префиксы REP, REPE, REPNE.
5. Постановка задачи
   1. Текст задания

В четвертой лабораторной работе необходимо ввести строку с клавиатуры, сделать ее обработку согласно заданию и показать результат на экране. При выполнении работы необходимо использовать хотя бы одну команду для работы с цепочками.

* 1. Условие задания

С клавиатуры вводится строка. Необходимо посчитать и вывести на экран, сколько раз в этой строке встречается каждая из согласных букв.

1. Программная реализация

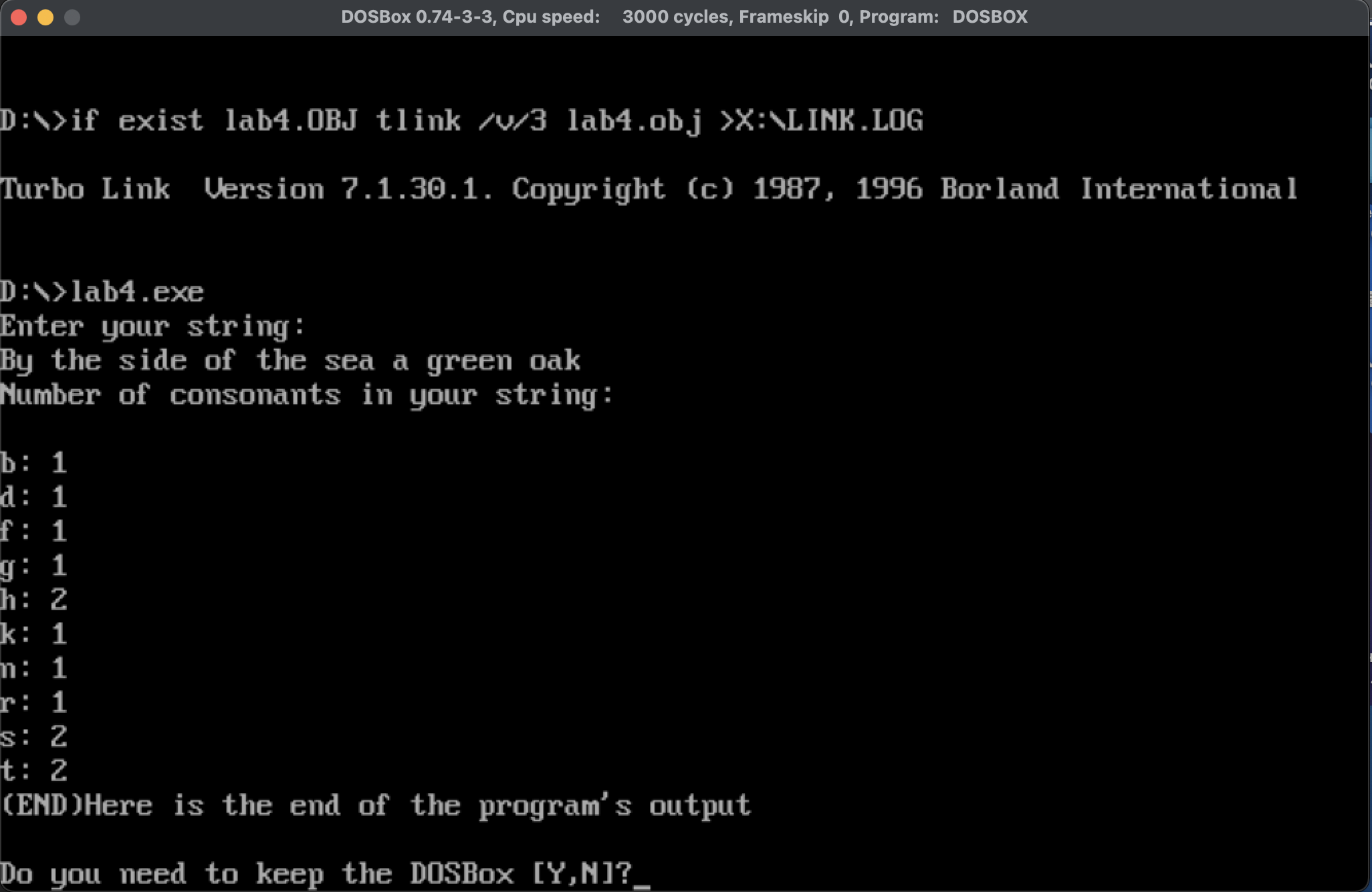
3.1 Обрабатываемая строка вводится с клавиатуры. Программа разбита при помощи меток на несколько частей, каждая из которых выполняет отвечает за определенную часть в обработке строки и выводе результата, согласно условию.

* 1. Результат можно видеть в выводе в консоли.
  2. Примеры:

3.3.1 Отладка программы №1

Проверка работы программы с обычной строкой.

Ввод: “By the side of the sea a green oak”

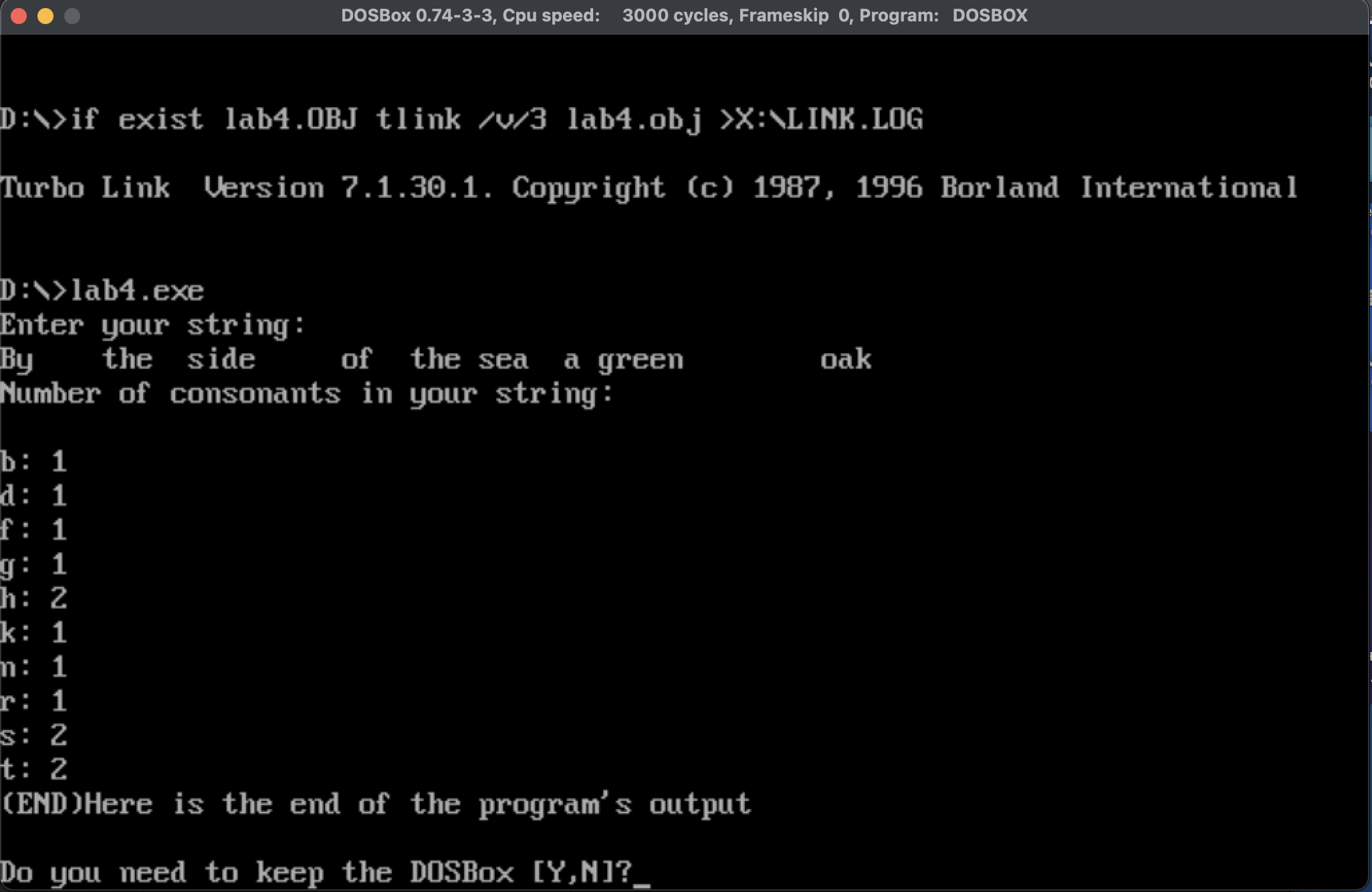


Результат: b – 1, d – 1, f – 1, g – 1, h – 2, k – 1, n – 1, r – 1, s – 2, t – 2.

3.3.2 Отладка программы №2

Проверка работы программы со строкой с дополнительными пробелами.

Ввод: “By the side of the sea a green oak”

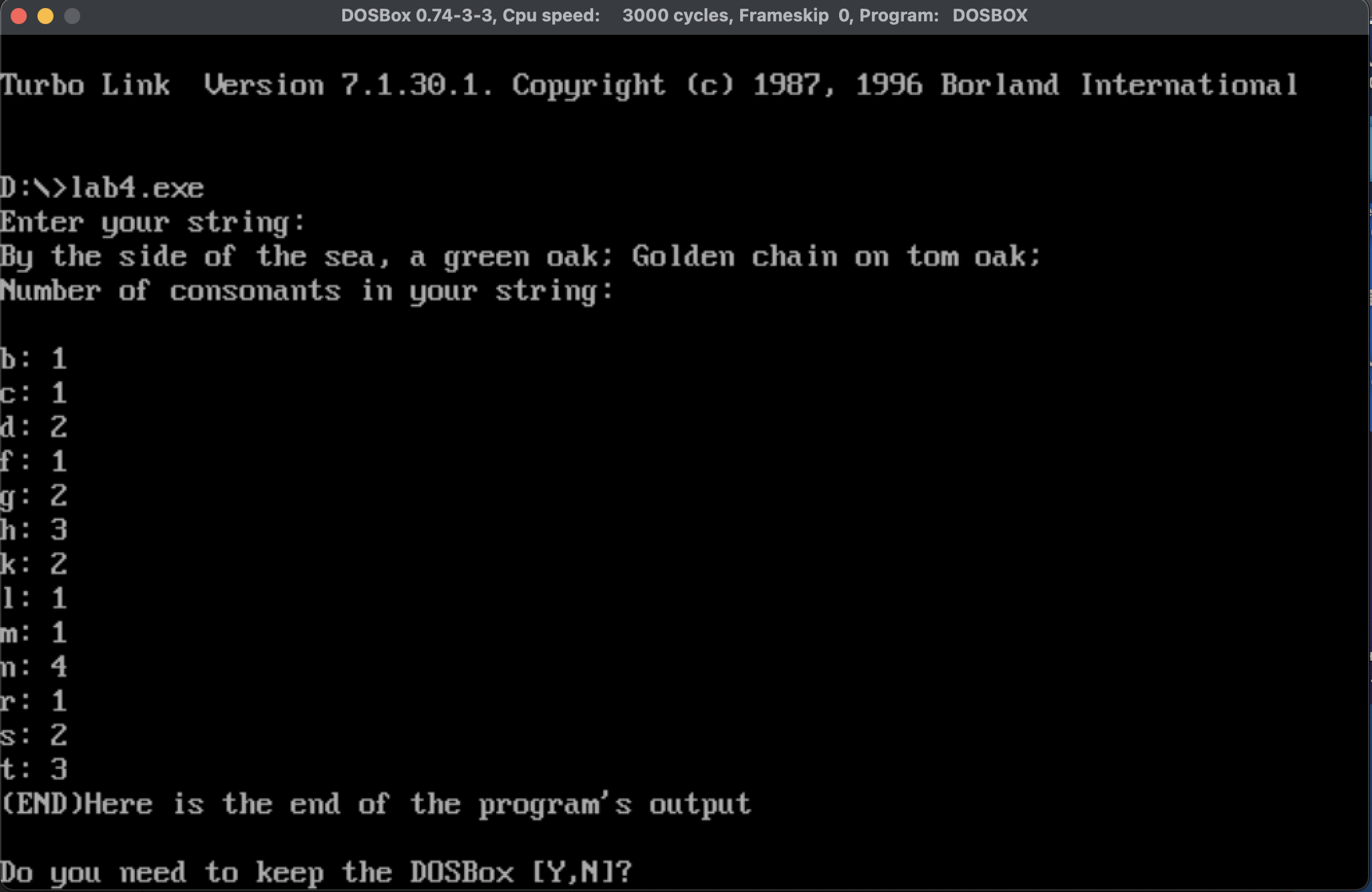


Результат: b – 1, d – 1, f – 1, g – 1, h – 2, k – 1, n – 1, r – 1, s – 2, t – 2.

3.4.3 Отладка программы №3

Проверка работы программы со строкой с дополнительными символами.

Ввод: “By the side of the sea, a green oak; Golden chain on tom oak;”

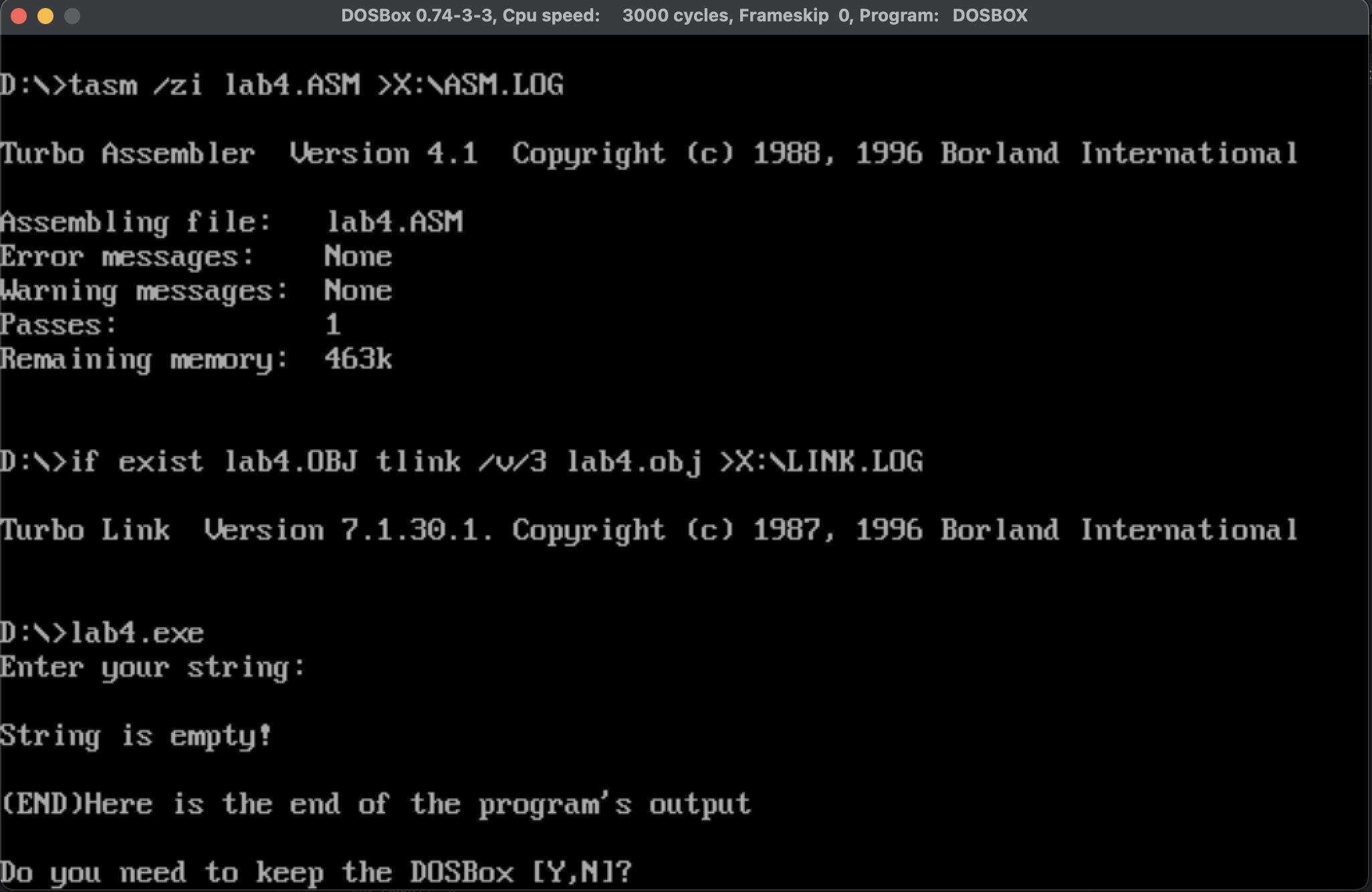


Результат: b – 1, c – 1, d – 2, f – 1, g – 2, h – 3, k – 2, l – 1, m – 1, n – 4, r – 1, s – 2, t – 3.

3.4.4 Отладка программы №4

Проверка работы программы с пустой строкой.

Ввод: “”



Результат: вывод об ошибке – пустая строка.

1. Выводы

На практике было изучено и опробовано, в соответствии с поставленной задачей: представление строки в языке Ассемблер, Команды movsb, movsw, stosb, stows, lodsb, lodsw, назначение флага направления, команды CLD и STD и префиксы REP, REPE, REPNE.

Литература

1. Юров В.И. – «Assembler. Учебник для вузов. 2-ое издание, 2003 год».

2. Юров В.И. – «Assembler. Практикум. 2-ое издание, 2006 год».

3. Калашников О.А. – «Ассемблер - это просто. 2-ое издание, 2011 год».

Приложение

.model small

.stack 100h

.data

string db 100 dup(?)

lowerCase db 'bcdfghjklmnpqrstvwxz$'

upperCase db 'BCDFGHJKLMNPQRSTVWXZ$'

startMessage db 'Enter your string:', 10, '$'

outputMessage db 'Number of consonants in your string:', 10, '$'

emptyException db 'String is empty!', 10, '$'

output db 0dh, 0ah, ' : $'

count dw 0

.code

start:

mov ax, @data

mov ds, ax

mov es, ax

startFunc:

mov ah, 9

lea dx, startMessage

int 21h

lea di, string

mov ah, 1

inputFunc:

int 21h

cmp al, 0dh ;проверка на нажатие ввод

je stopInput

mov [si], al

inc si

stosb

jmp inputFunc

stopInput:

mov al, "$" ;добавение $ в конец строки

stosb

cmp si, 0

je emptyString

mov bx, 0 ;номер в введенной строке

mov si, 0 ;номер в строках согласных

mov ah, 9

lea dx, outputMessage

int 21h

totalCount:

mov al, string[bx] ;помещение в al символа строки под номером bx

cmp al, "$"

je nextLetter

cmp al, lowerCase[si]

je isConsonant

cmp al, upperCase[si]

je isConsonant

inc bx

jmp totalCount

isConsonant:

mov cx, count

inc cx

mov count, cx

inc bx

jmp totalCount

nextLetter:

cmp count, 0

je skip ;если согласной нет - пропуск

mov ah, 9

mov dl, lowerCase[si]

mov output[2], dl

lea dx, output ;вывод результата по согласной

int 21h

mov ah, 2

mov dx, count

add dx, 30h

int 21h

skip:

inc si

cmp lowerCase[si], "$"

je exit

mov count, 0

mov bx, 0

jmp totalCount

emptyString:

mov ah, 9

lea dx, emptyException

int 21h

exit:

mov ah, 4ch

int 21h

end start