БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ИНО

Специальность ПОИТ

Индивидуальная практическая работа № 1

по дисциплине «Языки программирования»

часть 1

Вариант № 7

Выполнила: Н. Г. Карпеко

Обучение по сертификатам

Минск 2020

**Тема работы: Строковые команды.**

**Задание**: Определить, сколько элементов строки превышают код введенного символа (использовать команду scas). Строку описать в сегменте данных.

Файл iprone.asm содержит исходный код программы с реализацией задания. Компиляция, сборка и запуск программы выполнены в DOSBox Ver. 0.74-3.

В данной программе решены следующие задачи:

1. вывод строк «Stroka: » (переменная STRmsg) и «Vvedite simvol: » (переменная mesInput – запрос на ввод символа с клавиатуры, его код загружается в аккумулятор al);
2. вывод исходной строки «Primer 7» (переменная primerSTR);
3. сканирование строки (метка SkanStr) — сравниваем коды введенного символа и очередного элемента строки (команда scasb с префиксом повторения rep);
4. команда перехода jb выполняет переход к анализу флага после сравнения кодов символа и очередного элемента строки (метка IncNumber);
5. по завершению сканирования исходной строки (метка EndScan) происходит вывод результата (переменные mesOutput и Number).

В программе использованы две процедуры:

Процедура ShowMSG – производит вывод сообщения на экран. Процедура NewLine – переводит на новую строку на экране, используя переменную nextLine, описанную в сегменте данных.

Листинг программы:

; Программа определяет (считает) количество элементов строки, превышающих

; код введенного символа:

.model small ; модель памяти small (упрощенная директива сегментации)

.386 ; режим работы микропроцессора

.stack 256 ; сегмент стека (занимает 256 байтов)

.data ; сегмент данныхa

STRmsg db 'Stroka: ' ; первое сообщение, которое выводится на экран

primerSTR db 'Primer 7$' ; исходная строка

len = ($ - primerSTR) - 1 ; длина исходной строки

mesInput db 'Vvedite simvol: $' ; сообщение о вводе символа

mesOutput db 'Kolichestvo elementov bolshe koda vvedennogo simvola = $'

; сообщение результата

Number db 0 ; переменная для хранения результата

nextLine db 10, 13, '$' ; переменная для перевода на новую строчку

.code ; сегмент кода

main: ; метка начала программы

mov ax, @data ; занесение адреса сегмента данных в регистр ax

mov ds, ax ; копирование содержимого регистра ax

mov es, ax ; в регистры ds и es

**xor** ax, ax ; обнуление регистра ax

lea dx, STRmsg ; занесение в dx смещения переменной STRmsg

call ShowMSG ; вызов процедуры вывода исходной строки

call NewLine ; вызов процедуры перевода на новую строчку

lea dx, mesInput ; занесение в dx смещения переменной mesInput

call ShowMSG ; вызов процедуры вывода сообщения (mesInput)

mov ah, 01h ; занесение в ah функции 1

int 21h ; вызов прерывания

; ввод символа с клавиатуры и вывод его на экран, в регистр al загружается код

; символа:

mov di, offset primerSTR ; в di смещение исходной строки «Primer 7»

mov cx, len ; занесение в счетчик количества элементов строки «Primer 7»

cld ; сброс в 0 флага направления (DF)

SkanStr: ; метка начала сканирования строки «Primer 7» (primerSTR)

rep scasb ; сравнение введенного символа (в al) с исходной строкой

; primerSTR поэлементно

jb IncNumber ; если код введенного символа меньше очередного

; элемента исходной строки, то переход на метку IncNumber

jcxz EndScan ; если сканирование строки завершено (cx = 0), то переход

; на метку EndScan

IncNumber: ; метка увеличения кол-ва найденных элементов

setb bl ; занесения в регистр bl признака сравнения кодов введенного

; символа и очередного элемента строки

add Number, bl ; добавляем к переменной Number 1 - если код символа

; меньше, иначе – добавляем 0

jmp SkanStr ; переход к метке сканирования строки

EndScan: ; метка входа после окончания сканирования строки

call NewLine ; вызов процедуры перевода строки

lea dx, mesOutput ; занесение в dx смещения переменной mesOutput

call ShowMSG ; вызов процедуры вывода сообщения (mesOutput)

add Number, 30h ; добавляем число 30 к переменной Number для вывода

mov dx, offset Number ; занесение в dx смещения переменной Number

call ShowMSG ; вызов процедуры вывода результата (Number)

mov ah, 4ch ; завершение работы программы, возврат в DOS

int 21h ; вызов прерывания

; --------------- процедура перевода на новую строку NewLine: ----------------------

NewLine proc

mov dl, offset nextLine ; заносим в dl смещение переменной перевода строки

mov ah, 09h ; занесение в ah функции вывода строки (09h)

int 21h ; вызов прерывания, перевод строки

**xor** dl, dl ; обнуление используемых в процедуре

**xor** ah, ah ; регистров (dl и ah)

ret ; возврат из процедуры

NewLine endp ; --- конец процедуры NewLine

; ------------ процедура вывода сообщения на экран ShowMSG: ----------------------

ShowMSG proc

mov ah, 09h ; занесение в ah функции вывода строки (09h)

int 21h ; вызов прерывания, вывод информации

**xor** dx, dx ; обнуление используемых в процедуре

**xor** ah, ah ; регистров (dx, ah)

ret ; возврат из процедуры

ShowMSG endp ; --- конец процедуры ShowMSG

**end** main ; конец программы с точкой входа main.

Результаты запуска программы в DOSBox:

Тестовый набор данных: «A», «c», «7», «Z», «v», «u», «W». Данные символы будем вводить по очереди.

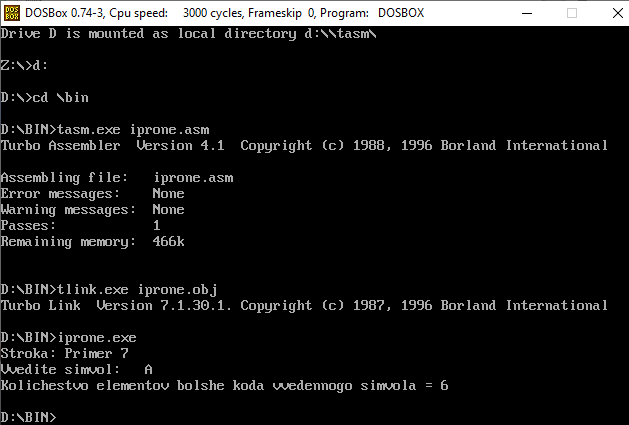


Рис. 1. Запуск программы – программа компилируется. Вводим символ «А», результат = 6 элементов строки превышают код введенного символа «А».

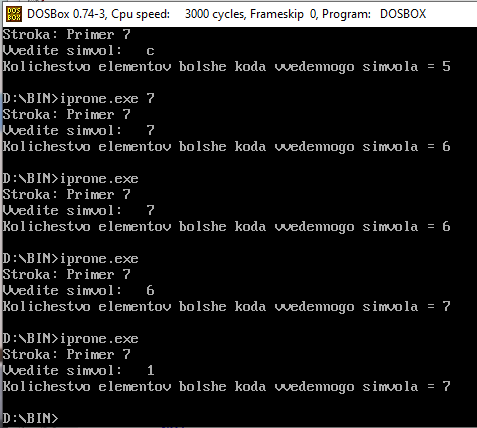


Рис. 2. Вводим символ «с», результат = 5.

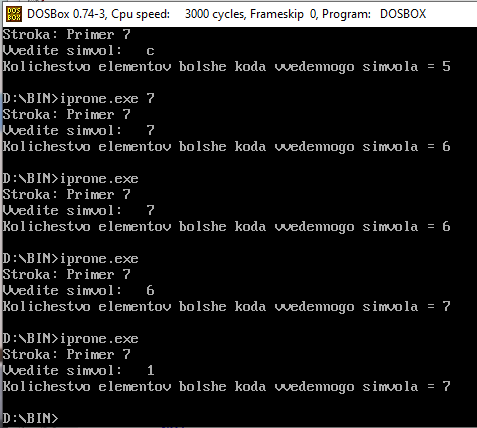


Рис. 3. Вводим символ «7», результат = 6.

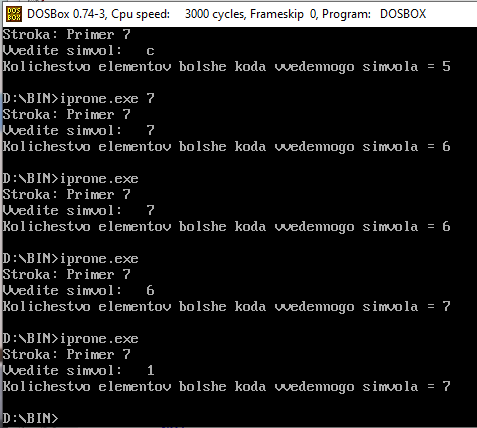


Рис. 4. Вводим символ «1», результат = 7.

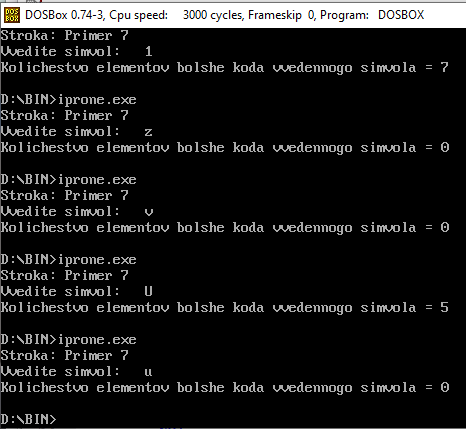


Рис. 5. Вводим символ «z», результат = 0.

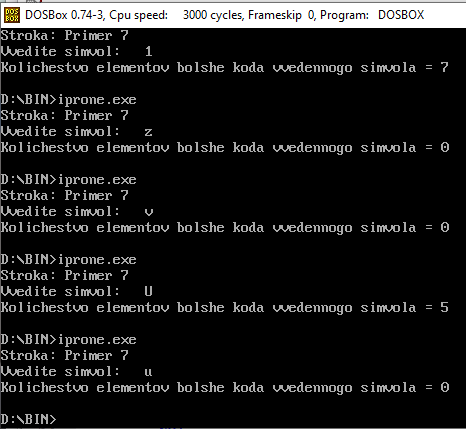


Рис. 6. Вводим символ «v», результат = 0.

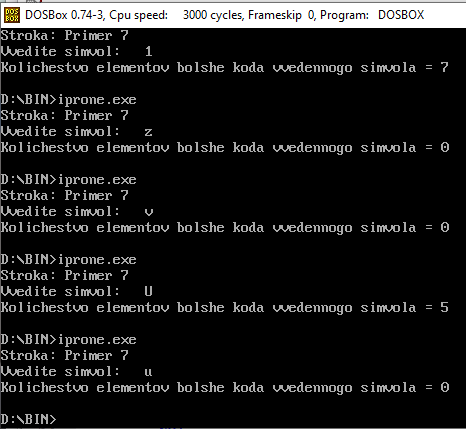


Рис. 7. Вводим символ «U», результат = 5.

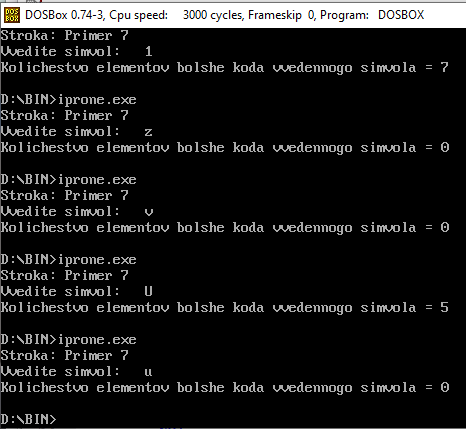


Рис. 8. Вводим символ «u», результат = 0.