ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной инженерии

Пояснительная записка к домашнему заданию

по дисциплине «Архитектура вычислительных систем»

Программа, которая на основе заданной ASCII-строки символов, решает вопрос, является ли данная строка палиндромом.

Исполнитель:

студент группы

БПИ193(1)

Полянская П. А.

02.11.2020 г.

Текст задания.

Разработать программу, которая на основе заданной ASCII-строки символов, решает вопрос, является ли данная строка палиндромом.

Используемые источники.

1) SoftCraft, сайт по учебной дисциплине. [Электронный ресурс]

http://softcraft.ru/ (дата обращения: 02.11.2020)

2) Ravesli, уроки по ассемблеру [Электронный ресурс]

[tps://ravesli.com/uroki-assemblera/](https://ravesli.com/uroki-assemblera/) (дата обращения: 02.11.2020)

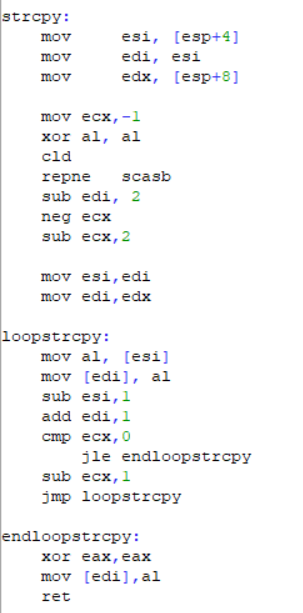
3) fasm - основное руководство [Электронный ресурс] <http://flatassembler.narod.ru/fasm.htm> (дата обращения: 02.11.2020)

Идея проверки строки ( является палиндромом или нет).

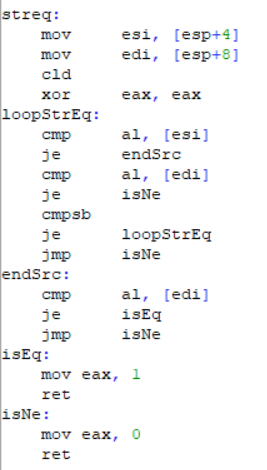
Проверяемую строку необходимо перевернуть и сравнить с исходной строкой, если они равны, то исходная строка является палиндромом, иначе - нет.

Назначения отдельных кусков кода.

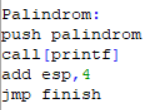
1) Данный код переворачивает исходную строку.



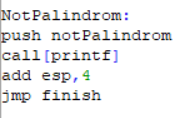
2) Данный код сравнивает исходную и перевернутую строку.



3) Если строка оказалась палиндромом, то происходит переход по метке Palindrom и выводится в консоль строка “This is palindrom!”.



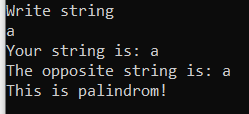
4) Если строка оказалась не палиндромом, то происходит переход по метке NotPalindrom и выводится в консоль строка “This is not a palindrom!”.



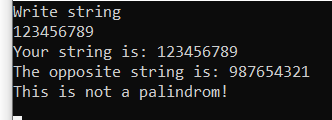
Тестирование различных входных данных.

В консоль вводится строка для проверки, а выводится сама строка, перевернутая строка и строка о том, является ли входная строка палиндромом.

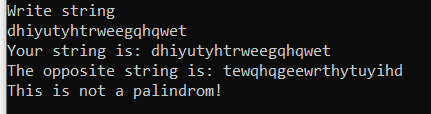
1) Вводится один символ. Он всегда является палиндромом.



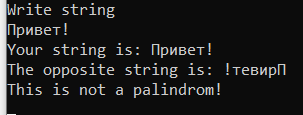
2) Вводится строка, не являющаяся палиндромом.



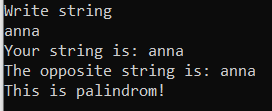
3) Вводится строка, не являющаяся палиндромом.



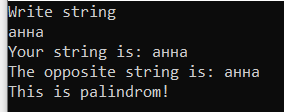
4) Вводится строка, не являющаяся палиндромом.



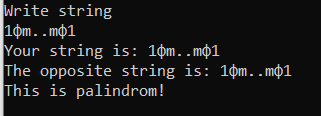
5) Вводится строка, являющаяся палиндромом.



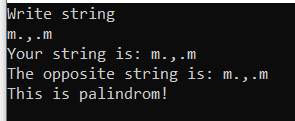
6) Вводится строка, являющая палиндромом.



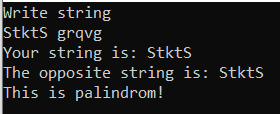
7) Вводится строка, являющая палиндромом.



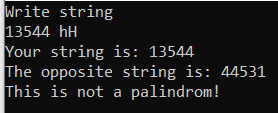
8) Вводится строка, являющая палиндромом.



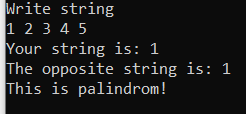
9) Если строка содержит больше одного слова, то рассматривается только первое слово.



10) Если строка содержит больше одного слова, то рассматривается только первое слово.



11) Если строка содержит больше одного слова, то рассматривается только первое слово.



Исходный код программы на языке ассемблера FASM.

Примечание: считаю, что вводится строка длиной не больше 255 символов.

;Полянская Полина БПИ193 1 подгруппа

;Разработать программу, которая на основе заданной ASCII-строки символов, решает вопрос, является ли данная строка палиндромом

;17 вариант

format PE console

entry start

include 'win32a.inc'

section '.data' data readable writable

input db 'Write string',10,0

; строка, чтобы вывести перевернутую строку

formatStr db '%s', 0 ; для считывания строки с консоли

thisString db 'Your string is: %s',10,0

; строка, чтобы вывести исходную строку

oppositeString db 'The opposite string is: %s',10,0

; строка, чтобы вывести перевернутую строку

palindrom db 'This is palindrom!',10,0

; строка, чтобы сообщить о том, что исходная строка - палиндром

notPalindrom db 'This is not a palindrom!',10,0

; строка, чтобы сообщить о том, что исходная строка - не палиндром

nm rb 256 ; исходная строка

nm2 rb 256 ; перевернутая строка

;--------------------------------------------------------------------------

; Main program

section '.code' code readable executable

start:

push input

call [printf]

add esp, 4

push nm ;считываем строку с консоли

push formatStr

call[scanf]

add esp, 8

push nm ; выводим исходную строку

push thisString

call [printf]

add esp, 8

stdcall strcpy, nm, nm2 ; переворачиваем исходную строку

push nm2

push oppositeString

call [printf] ; выводим перевернутую строку

stdcall streq, nm,nm2

; сравниваем строки

cmp eax, 0 ;если значение в eax = 0, то переходим по метке NotPalindrom

je NotPalindrom

cmp eax,1 ; если значение в eax =1, то переходим по метке Palindrom

je Palindrom

;--------------------------------------------------------------------------

Palindrom:

push palindrom

call[printf] ;выводим строку, о том, что строка - палиндром

add esp,4

jmp finish ; переходим на метку finish

NotPalindrom:

push notPalindrom

call[printf] ; выводим строку, о том, что строка - не палиндром

add esp,4

jmp finish ; переходим на метку finish

finish:

call [getch] ; при введении пользователем символа - завершаем программу

push 0

call [ExitProcess]

strcpy:

mov esi, [esp+4]

mov edi, esi

mov edx, [esp+8]

mov ecx,-1

xor al, al ; al=0

cld ; направление от начала до конца

repne scasb

sub edi, 2

neg ecx

sub ecx,2

mov esi,edi

mov edi,edx

loopstrcpy:

mov al, [esi]

mov [edi], al

sub esi,1

add edi,1

cmp ecx,0

jle endloopstrcpy ; если значение в ecx меньше или равно 0, то переходим по метке ;endloopstrcpy

sub ecx,1

jmp loopstrcpy ; переходим по метке loopstrcpy

endloopstrcpy:

xor eax,eax ; eax=0

mov [edi],al

ret ; возвращаемся

streq:

mov esi, [esp+4]

mov edi, [esp+8]

cld ; направление от начала до конца

xor eax, eax ; eax=0

loopStrEq:

cmp al, [esi]

je endSrc ; если al=[esi], то переходим по метке endSrc

cmp al, [edi]

je isNe ; если al=[edi], то переходим по метке isNe

cmpsb

je loopStrEq

jmp isNe ; переходим по метке isNe

endSrc:

cmp al, [edi]

je isEq ;если al=[edi], то переходим по метке isEq

jmp isNe ; переходим по метке isNe

isEq:

mov eax, 1 ; перемещаем в eax 1

ret ; возвращаемся

isNe:

mov eax, 0 ; перемещаем в eax 0

ret ; возвращаемся

;--------------------------------------------------------------------------

; Including External Api

section '.idata' import data readable

library kernel, 'kernel32.dll',\

msvcrt, 'msvcrt.dll'

include 'api\kernel32.inc'

import kernel,\

ExitProcess, 'ExitProcess'

import msvcrt,\

printf, 'printf',\

scanf, 'scanf',\

getch, '\_getch'