Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина «Конструирование программ»

**ОТЧЕТ**

к лабораторной работе №4

на тему:

**«ЦЕЛОЧИСЛЕННЫЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ. ОБРАБОТКА МАССИВОВ ЧИСЛОВЫХ ДАННЫХ»**

БГУИР 6-05-0612-02 001

|  |
| --- |
| Выполнил студент группы 353503  АБДУЛОВ Александр Алексеевич |
|  |
| (дата, подпись студента) |
| Проверил ассистент каф. Информатики  РОМАНЮК Максим Валерьевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 Индивидуальное задание

**Задание 1. Вариант 1.** Перевести число из одной системы счисления в другую (системы счисления задаются до ввода-вывода числа в диапазоне [2..16]).

# 2 Выполнение работы

Программа для перевода чисел из одной системы счисления в другую. Поддерживая базы от 2 до 16, запрашивается у пользователя начальная база, число в этой базе и новую базу для перевода. В коде используются директивы .model, .stack, .data и .code для определения структуры программы. В секции данных объявляются сообщения для ввода и вывода, а также переменные для хранения введенных значений и промежуточных результатов.

.model small

.stack 100h

.data

entrInBsStr db "Enter initial base (2-16):", '$'

entrInNumStr db 0Dh, 0Ah, "Enter number (no more 0-65535 in decimal) in initial base:", '$'

entrNewBsStr db 0Dh, 0Ah, "Enter new base (2-16):", '$'

trnslStr db 0Dh, 0Ah, "Translated number:", '$'

errMSG db 0Dh, 0Ah ,"ERROR, WRONG INPUT!", 0Dh, 0Ah, '$'

simBssStr db 0Dh, 0Ah, "BASES ARE EQUAL!", '$'

tmp db 200,?,200 dup('$')

inNum dw 0h

inBase dw 0h

finBase dw 0h

finNum dw 0h

isNegative dw 0h

.code

;;;;;;;;;;;;;;;;;;

print macro string

lea dx, string

mov ah, 09h

int 21h

endm

;;;;;;;;;;;;;;;;;

input macro string

lea dx, string

mov ah, 0Ah

int 21h

endm

;;;;;;;;;;;;;;;;;;;;;;;;;

main:

mov ax, @data

mov ds, ax

print entrInBsStr

input tmp

xor ax, ax

mov al,tmp[1]

cmp al,0

je error ;empty str

push ax

push 10

lea di, tmp+2

call atoi

jo error

cmp dx, 0xFFFFh

je error

cmp dx, 0xEEEEh

je error

cmp dx, 1h

je error

mov inBase, ax

push inBase

call checkBase

cmp dx, 0h

je error ;illegal base

print entrInNumStr

input tmp

xor ax, ax

mov al,tmp[1]

cmp al,0

je error ;empty srt

push ax

push inBase

lea di, tmp+2

call atoi

jo error ;overflow

cmp dx, 0xFFFFh

je error ;illegal char

cmp dx, 0xEEEEh

je error ;illegal num

mov inNum, ax

cmp dx, 1h

jne signConfirmed

inc isNegative

signConfirmed:

print entrNewBsStr

input tmp

xor ax, ax

mov al,tmp[1]

cmp al,0

je error ;empty srt

push ax

push 10

lea di, tmp+2

call atoi

jo error ;overflow

cmp dx, 0xFFFFh

je error ;illegal char

cmp dx, 0xEEEEh

je error ;illegal num

cmp dx, 1h

je error ;illegal base init

mov finBase, ax

push finBase

call checkBase

cmp dx, 0h

je error ;illegal base new

mov ax, inBase

cmp ax, finBase

je simBase

push inNum

push finBase

lea di, tmp+2

call translate

print trnslStr

push isNegative

push finBase

call printInRevOrder

jmp endProg

error:

print errMSG

jmp endProg

simBase:

print simBssStr

jmp endProg

;;;;;;;;;;;;;;;;

atoi proc near

;stack contains: base

;puts sign in dx

push bp

mov bp, sp

mov bx, [bp+4] ;base

mov cx, ax ;length

xor dx, dx

sign dw 0h

tmpVal dw 0h

mov sign, dx ;to clear sign field

mov tmpVal, dx ;to clear sign field

;sign check

mov dl, [di]

mov ax, dx

cmp ax, '+'

je signConf

cmp ax, '-'

jne start

inc sign

signConf:

inc di ;next char

dec cx

start:

xor ax, ax ;init

xor dx, dx

next1:

mul bx

jo finish1

mov dl, [di]

mov tmpVal, ax

mov ax, dx

lowCheck:

cmp al, 'f'

ja charLocated

cmp al, 'a'

jb highCheck

sub dl, 57h

jmp checksPassed

highCheck:

cmp al, 'F'

ja charLocated

cmp al, 'A'

jb numCheck

sub dl, 37h

jmp checksPassed

numCheck:

cmp al, '9'

ja charLocated

cmp al, '0'

jb charLocated

sub dl, 30h

checksPassed:

cmp bl, dl

jbe wrongBase

possOvflwCheck:

mov ax, tmpVal

cmp ax, 0xFFF1h

jbe noOverflow

add ax, dx

cmp ax, 3h

ja noOverflow

mov ax, tmpVal

mul bx

jo finish1

noOverflow:

mov ax, tmpVal

add ax, dx

jo finish1

inc di

loop next1

mov dx, sign

jmp finish1

wrongBase:

mov dx, 0xEEEEh

jmp finish1

charLocated:

mov dx, 0xFFFFh

finish1:

pop bp

ret

atoi endp

;;;;;;;;;;;;;;;;

checkBase proc near

;stack contains: base for checking

push bp

mov bp, sp

mov bx, [bp+4]

mov ax, 2

cmp bx, ax

jl wrong2

mov ax, 16

cmp bx, ax

ja wrong2

mov dx, 0x1h

jmp finish2

wrong2:

xor dx, dx

finish2:

pop bp

ret

checkBase endp

;;;;;;;;;;;;;;;;

translate proc near

; stack contains: new base, initial number

push bp

mov bp, sp

mov bx, [bp+4] ;base

mov ax, [bp+6] ;number

xor cx, cx

dec di

again:

inc di

xor dx, dx

div bx

mov [di], dl

inc cx

cmp ax, 0

ja again

pop bp

ret

translate endp

printInRevOrder proc near

push bp

mov bp, sp

mov bx, [bp+4]

mov ax, [bp+6]

cmp ax, 0h

je next2

mov dx, '-'

inc di

inc cx

jmp lowerThan9

next2:

mov dl, [di]

add dx, 30h

cmp dx, 39h

jle lowerThan9

add dx,7

lowerThan9:

mov ah, 2

int 21h

dec di

loop next2

pop bp

ret

printInRevOrder endp

endProg:

mov ax, 4C00h

int 21h

;;;;;;;;;;;;;;;;;;;;;;

end main

В коде программы применяется макрос для упрощения вывода строковых сообщений на экран. Процедура atoi преобразует строку символов в целое число, включая проверку знака и контроль за переполнением. Процедура checkBase проверяет корректность введенной базы, гарантируя, что она находится в пределах от 2 до 16. Процедура translate выполняет преобразование числа из одной системы счисления в другую, а printInRevOrder выводит результат в обратном порядке, что необходимо для отображения чисел в новой системе.

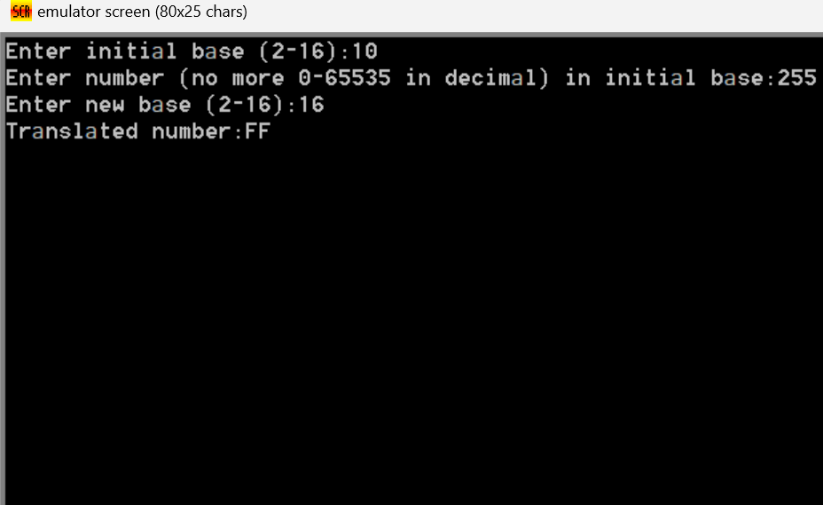


Рисунок 1 – Результат выполнения программы

# Вывод

В ходе лабораторной работы была изучена работа с целочисленными арифметическими операциями и обработка массивов, а также работа с процедурами. Основное внимание уделялось операциям сложения, вычитания, умножения.