

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 №5

на тему:

**«Целочисленные арифметические операции. Обработка массивов
числовых данных»**

БГУИР 1-40 04 01

Выполнил студент группы 253505

Снежко Максим Андреевич

(дата, подпись студента)

Проверил ассистент кафедры
информатики

Романюк Максим Валерьевич

(дата, подпись преподавателя)

Минск 2023

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

Ход работы: На рисунке 1 представлен ввод системы счисления, затем число в десятичной системе счисления. И конвертация заданного числа в заданную систему счисления

Листинг 1 – Исходный код программы задания 1

```
.model small
.stack 100h
.data
    inputLimitsString db "Number system - [2,16]. ",
0Dh, 0Ah, "Min value = -32768, max value = 32767.$"
    notation db "Enter the number system: $"
    inputInviteString db "Enter number: $"
    invalidLengthString db "Invalid length input. 2 <=
length <= 16.$"
    invalidInputString db "Invalid input. $"
    overflowInputString db "Number is too big. $"
    tryAgainString db "Try again: $"
    newLine db 0Dh,0Ah,'$'
    length dw ?
    number dw ?
    sign db ?
    output_msg db "Converted number: $"
    yes dw ?
    numberTen dw 000Ah
    sizeOfNumber equ 2
    numberStringLength equ 20
    numberString db numberStringLength dup('$')
    askRestartString db "Do you want to restart
program? 1=yes, other-no: $"
    maxArrayLength equ 20h ;16
    array dw maxArrayLength dup(?)
.code
inputNumbers proc
    call printNewLine
    lea dx, inputInviteString
    call outputString
repeatElementInput:
    lea dx, numberString
    call inputString
    lea si, numberString[2]
    call parseString
    jc invalidInput
```

```

        call loadNumber
        loop inputNumbers
ret
invalidInput:
    call printNewLine
    lea dx, invalidInputString
    call outputString
    jno tryAgainOutput
    lea dx, overflowInputString
    call outputString
tryAgainOutput:
    lea dx, tryAgainString
    call outputString
    jmp repeatElementInput
loadNumber:
    mov [di], ax
    add di, sizeofNumber
ret
inputNumbers endp
parseString proc
    xor dx, dx
    xor bx, bx
    xor ax, ax
    jmp inHaveSign
parseStringLoop:
    mov bl, [si]
    jmp isNumber
validString:
    sub bl, '0'
    imul numberTen
    jo invalidString
    js invalidString
    add ax, bx
    js invalidString
    inc si
    jmp parseStringLoop
isNumber:
    cmp bl, 0Dh
    je endParsing
    cmp bl, '0'
    jl invalidString
    cmp bl, '9'
    jg invalidString
    jmp validString
inHaveSign:

```

```

        cmp [si], '-'
        je negative
        ;push 1
        mov sign, '+'
        cmp [si], '+'
        jne isNullString
        inc si
        jmp isNullString
negative:
        mov sign, '-'
        inc si
        jmp isNullString
isNullString:
        cmp [si], 0Dh
        je invalidString
        jmp parseStringLoop
invalidString:
        cmp ax, 32768
        je goqwe
inv:
        stc
ret
endParsing:
        clc
ret
setMinusSign:
        mov sign, '-'
        clc
ret
goqwe:
        cmp sign, '-'
        jne inv
        clc
ret
parseString endp
printNewLine proc
        lea dx, newLine
        call outputString
ret
printNewLine endp
outputString proc
        mov ah, 09h
        int 21h
ret
outputString endp

```

```

inputString proc
    mov ah, 0Ah
    int 21h
ret
inputString endp
NumberSystemInput proc
    call printNewLine
    lea dx, notation
    call outputString
    lea di, length
    mov cx, 0001h
    call inputNumbers
    cmp ax, maxArrayLength
    jg invalidLengthInput
    cmp ax, 0002h
    jl invalidLengthInput
    mov al, [sign]
    cmp al, '-'
    je invalidLengthInput
ret
invalidLengthInput:
    call printNewLine
    lea dx, invalidLengthString
    mov sign, '+'
    call outputString
    call printNewLine
    jmp NumberSystemInput
NumberSystemInput endp
NumberInput proc
    call printNewLine
    lea di, number
    mov cx, 0001h
    call inputNumbers
    mov number, ax
    call printNewLine
ret
NumberInput endp
decimalToCustomSystem proc
    lea dx, output_msg
    call outputString
    mov ax, number
    cmp ax, 0
    je nol
    mov al, [sign]
    cmp al, '-'

```

```

        jne notMinus
        mov dl, '-'
        mov ah, 2
        int 21h
notMinus:
        mov ax, number
        xor cx, cx
        xor dx, dx
        mov bx, [length]
        cmp ax, 0h
        je zeroResult
        cmp ax, 0
        jns conversionLoop
        jz zeroResult
conversionLoop:
        mov dx, 0
        div bx
        push dx
        inc cx
        cmp ax, 0
        jz zeroResult
        cmp ax, 0h
        je zeroResult
        cmp ax, 0
        jns conversionLoop
        jz zeroResult
zeroResult:
        mov ah, 2
printLoop:
        pop dx
        add dl, '0'
        cmp dx, '9'
        jbe notGreaterThan9
        add dl, 7
notGreaterThan9:
        int 21h
        loop printLoop
        ret
nol:
        mov ah, 2
        mov dx, ax
        add dl, '0'
        int 21h
ret
decimalToCustomSystem endp


```

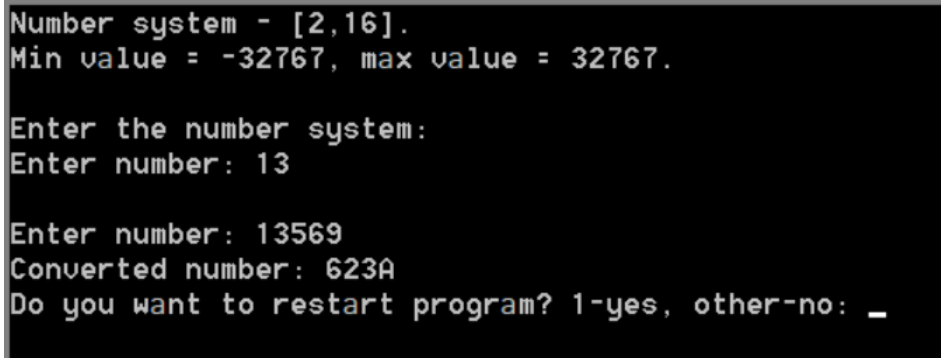
```

AskToRestart proc
    call printNewLine
    lea dx, askRestartString
    call outputString
    mov ah, 01h
    int 21h
    ret
AskToRestart endp
ClearConsole proc
    mov ah, 06h
    mov al, 0
    mov bh, 07h
    mov ch, 0
    mov cl, 0
    mov dh, 24
    mov dl, 79
    int 10h
    mov ah, 02h
    mov bh, 0
    mov dh, 0
    mov dl, 0
    int 10h
    ret
ClearConsole endp
start:
    mov ax, @data
    mov ds, ax
    mov es, ax,
    xor ax, ax
    lea dx, inputLimitsString
    call outputString
    call printNewLine
    call NumberSystemInput
    mov ax, maxArrayLength[1]
    call NumberInput
    call decimalToCustomSystem
    call AskToRestart
    cmp al, '1'
    je restartProgram
    jmp programEnd
restartProgram:
    call ClearConsole
    mov sign, 0
    jmp start
programEnd:

```

```
mov ah, 4Ch  
int 21h  
end start
```

 emulator screen (80x25 chars)



```
Number system - [2,16].  
Min value = -32767, max value = 32767.  
  
Enter the number system:  
Enter number: 13  
  
Enter number: 13569  
Converted number: 623A  
Do you want to restart program? 1=yes, other-no: _
```

Рисунок 1 – выполнение программы

Вывод: выполнено задание перевести число из одной системы счисления в другую (системы счисления задаются до ввода-вывода числа в диапазоне [2..16]). Таким образом были изучены основные целочисленные арифметические операции, обработка массивов числовых данных, логические побитовые операции над целыми числами, а также работа с процедурами для работы на ассемблере для процессора Intel 8086