

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 253504
ЖГУТОВ Евгений Дмитриевич

(дата, подпись студента)

Проверил ассистент кафедры информатики
РОМАНЮК Максим Валерьевич

(дата, подпись преподавателя)

Минск 2023

Цель работы: Задание 1. Вариант 7. Ввести массив целых чисел размерностью 30 элементов. Найти наиболее часто встречающееся число.

Ход работы: На рисунке 1 – результат работы программы.

Листинг 1 – Исходный код программы задания 1

```
.model small
.stack 100h

.data
    errorMessage      db 'ERROR$'
    greetingMessage    db 'Please enter the amount
of numbers to input:$'
    inputNumberMessage db 'Enter a number from -
32768 to 32767:$'
    repeatInputMessage db 'Repeat input!!!$'
    resultMessage      db 'Result:$'
    NewLine            db 10, 13, '$'
    numbersArray        dw 30 dup (?)
    inputBuffer         db 7,8 dup(?)
    repeatCount1        db ?
    repeatCount2        db ?
    arrayLength         dw ?
    sign                db '-', '$'
    tempNumber          dw ?

.code
inputNumber proc near
    mov ah, 0ah
    mov dx, offset inputBuffer
    int 21h
    ret
inputNumber endp

convertToInteger proc near
    xor ax, ax
    xor dx, dx
    lodsb
    push ax
    cmp ax, '-'
    jne skipMinusCheck
    dec cx
    lodsb
```

```

skipMinusCheck:
    cmp ax, '0'
    jb invalidCharacter
    cmp ax, '9'
    ja invalidCharacter
    sub ax, '0'
    push ax
    mov ax, numbersArray[di]
    jo overflowError
    mul bx
    jo overflowError
    mov numbersArray[di], ax
    jo overflowError
    pop ax
    add numbersArray[di], ax
    jo invalidCharacter
    lodsb
    loop skipMinusCheck
    pop ax
    cmp ax, '-'
    je negativeConversion
    jmp endConvert

```

```

invalidCharacter:
    pop ax
    cmp ax, '-'
    je negativeConversion

```

```

errorInvalid:
    mov ah, 09h
    lea dx, NewLine
    int 21h
    mov ah, 09h
    lea dx, errorMessage
    int 21h
    mov dx, 1
    jmp endConvert

```

```

overflowError:
    pop ax
    pop ax
    mov ah, 09h
    lea dx, NewLine
    int 21h
    mov ah, 09h

```

```

        lea dx, errorMessage
        int 21h
        mov dx, 1
        jmp endConvert

negativeConversion:
        not numbersArray[di]
        add numbersArray[di], 1
        cmp numbersArray[di], -32768
        js errorInvalid
        jmp endConvert
endConvert:
        ret
convertToInteger endp

searchDuplicates proc near
        mov repeatCount1, 0
        mov cx, arrayLength

searchLoop:
        xor ax, ax
        mov ax, numbersArray[si]
        repne scasw

        jne foundUniqueNumber

        inc repeatCount1
        mov dl, repeatCount1
        jcxz foundUniqueNumber
        jmp searchLoop
foundUniqueNumber:
        inc si
        inc si
        ret

searchDuplicates endp

DecToASCII proc near
@b:
        div bx
        add dx, '0'
        push dx
        inc cx
        xor dx, dx
        cmp ax, 0

```

```

        jne @b
@a:
    pop bx
    mov inputBuffer[si+2], bl
    inc si
    loop @a
    mov inputBuffer[si+2], '$'
    ret
DecToASCII endp

```

```

main:
    mov ax, @data
    mov ds, ax
    mov es, ax

```

```

start:
    mov ah, 09h
    ;lea dx, NewLine
    ;int 21h
    mov numbersArray[di], 0
    mov di, 0
    mov ah, 09h
    lea dx, greetingMessage
    int 21h
    call inputNumber
    lea si, inputBuffer+2
    mov bx, 10
    xor cx, cx
    mov cl, inputBuffer[1]
    call convertToInteger
    add cx, numbersArray[di]
    js start
    xor cx, cx
    mov ax, numbersArray[di]
    mov numbersArray[di], 0
    mov arrayLength, ax
    mov bx, 30
    cmp bx, arrayLength
    jl start
    mov bx, 0
    cmp bx, arrayLength
    je start
    mov cx, arrayLength
    mov di, 0
    mov repeatCount2, 0

```

```

        jmp main1

repeatInput:
    mov ah, 09h
    lea dx, NewLine
    int 21h
    mov ah, 09h
    lea dx, repeatInputMessage
    int 21h
    dec di
    dec di
    mov numbersArray[di], 0

main1:
    push cx
    mov ah, 09h
    lea dx, NewLine
    int 21h
    mov ah, 09h
    lea dx, inputNumberMessage
    int 21h
    call inputNumber
    lea si, inputBuffer+2
    mov bx, 10
    mov cl, inputBuffer[1]
    call convertToInteger
    pop cx
    inc di
    inc di
    cmp dx, 1
    je repeatInput
loop main1
    mov cx, arrayLength
    mov repeatCount1, 0
    xor si, si

RepeatedValuesSearch:
    push cx
    mov di, offset numbersArray
    call searchDuplicates
    mov bl, repeatCount2
    mov bh, repeatCount1
    cmp bh, bl
    jg swapCounts
    pop cx

```

```

        loop RepeatedValuesSearch
        jcxz exitProgram

swapCounts:
        mov bl, bh
        mov repeatCount2, bl
        pop cx
        mov tempNumber, ax
        loop RepeatedValuesSearch
        jcxz exitProgram

exitProgram:
        mov ah, 09h
        lea dx, NewLine
        int 21h
        mov ah, 09h
        lea dx, resultMessage
        int 21h
        xor ax, ax
        add ax, tempNumber
        js processNegativeNumber
        jns printResult

processNegativeNumber:
        not ax
        add ax, 1
        xor cx, cx
        mov cx, ax
        mov ah, 09h
        lea dx, sign
        int 21h
        mov ax, cx

printResult:
        mov bx, 10
        xor si, si
        xor dx, dx
        xor cx, cx
        call DecToASCII
        mov ah, 09h
        lea dx, inputBuffer+2
        int 21h
        mov ah, 4ch
        int 21h
end main

```

```
5ch emulator screen (80x25 chars)

Please enter the amount of numbers to input:reh
ERRORPlease enter the amount of numbers to input:$
ERRORPlease enter the amount of numbers to input:/
ERRORPlease enter the amount of numbers to input:reh235
ERRORPlease enter the amount of numbers to input:6
Enter a number from -32768 to 32767:1
Enter a number from -32768 to 32767:1
Enter a number from -32768 to 32767:1
Enter a number from -32768 to 32767:1
Enter a number from -32768 to 32767:2
Enter a number from -32768 to 32767:2
Result:1
```

Рисунок 1 – Результат работы программы

Выводы: В результате лабораторной работы была выполнена одна задача с использованием целочисленных арифметических операций и обработки массивов числовых данных.