**Комп’ютерний практикум №2**

**Тема:** засоби обміну даними

**Завдання**:

1. Написати процедуру введення і перетворення цілого числа.
2. Виконати математичну дію над числом (номер завдання вибирати за останніми двома числами номеру в заліковій книжці).
3. Перевести число в рядок та вивести його на екран.

**Варіант** №6: +78

**Код програми**:

STSEG SEGMENT PARA STACK "STACK"

    DB 64 DUP (0)

STSEG ENDS

DSEG SEGMENT PARA PUBLIC "DATA"

    buffError db "Number is too big. From: -32768 to 32767$"

    startStr db "Enter a number : $"

    inputError db "Number has incorrect chars.", 10, "$"

    bufferSize DB 7     ; 6 chars + RETURN

    inputLength DB 0        ; number of read chars

    buffer DB 7 DUP('?')    ; actual buffer

DSEG ENDS

CSEG SEGMENT PARA PUBLIC "CODE"

    MAIN PROC FAR

        ASSUME cs: CSEG, ds: DSEG, ss:STSEG

        push ds

        xor ax, ax

        PUSH ax

        MOV ax,DSEG

        MOV ds, ax

        xor dx, dx

        lea dx, startStr

        call WRITING

        call READING

        mov al, 10

        int 29h

        call ATOI

        jo overflow\_error

        add ax, 78

        jo overflow\_error

        mov bx, ax

        call ITOA

        jmp end\_main

        overflow\_error:

            xor dx, dx

            lea dx, buffError

            call WRITING

        end\_main:

            ret

    MAIN ENDP

    ITOA PROC NEAR

        or bx, bx

        jns positive\_number

        xor ax, ax

        mov al, '-'

        int 29h

        neg bx

        positive\_number:

            mov ax, bx

            xor cx, cx  ; chars number

            mov bx, 10  ; diviator

            itoa\_loop:

                xor dx, dx  ; remainder stores here

                div bx

                add dl, '0'

                push dx

                inc cx

                test ax, ax

                jnz itoa\_loop

            output:

                pop ax

                int 29h

                loop output

            ret

        ITOA ENDP

    ATOI PROC NEAR

        xor ax, ax      ; result

        xor bx, bx      ; iterator

        lea bx, buffer

        xor cx, cx      ; char

        xor di, di      ; 10

        mov di, 10

        xor si, si      ; sign

        skip\_whitespaces:

            mov cl, BYTE PTR [bx]

            cmp cl, 32  ; ' '

            je step

            cmp cl , 9  ; '\t'

            je step

            jmp sign\_check

        step:

            inc bx

            jmp skip\_whitespaces

        sign\_check:

            cmp cl, 45  ; '-'

            je minus

            cmp cl, 43  ; '+'

            je plus

            jmp atoi\_loop

        minus:

            mov si, 1

            plus:

                inc bx

                jmp atoi\_loop

        atoi\_loop:

            mov cl, BYTE PTR [bx]

            cmp cl, 48  ; '0'

            jl atoi\_end

            cmp cl, 57  ; '9'

            jg atoi\_end

            imul di

            jo error

            sub cl, 48

            add ax, cx

            jo exit\_atoi

            ;jo error

            inc bx

            jmp atoi\_loop

        atoi\_end:

            cmp si, 1

            je make\_neg

            jmp exit\_atoi

        make\_neg:

            neg ax

            jmp exit\_atoi

        exit\_atoi:

            ret

        ATOI ENDP

    READING PROC NEAR

        lea dx, bufferSize

        mov ah, 10

        int 21h

        ret

    READING ENDP

    WRITING PROC NEAR

        mov ah, 9

        int 21h

        ret

    WRITING ENDP

CSEG ENDS

END MAIN

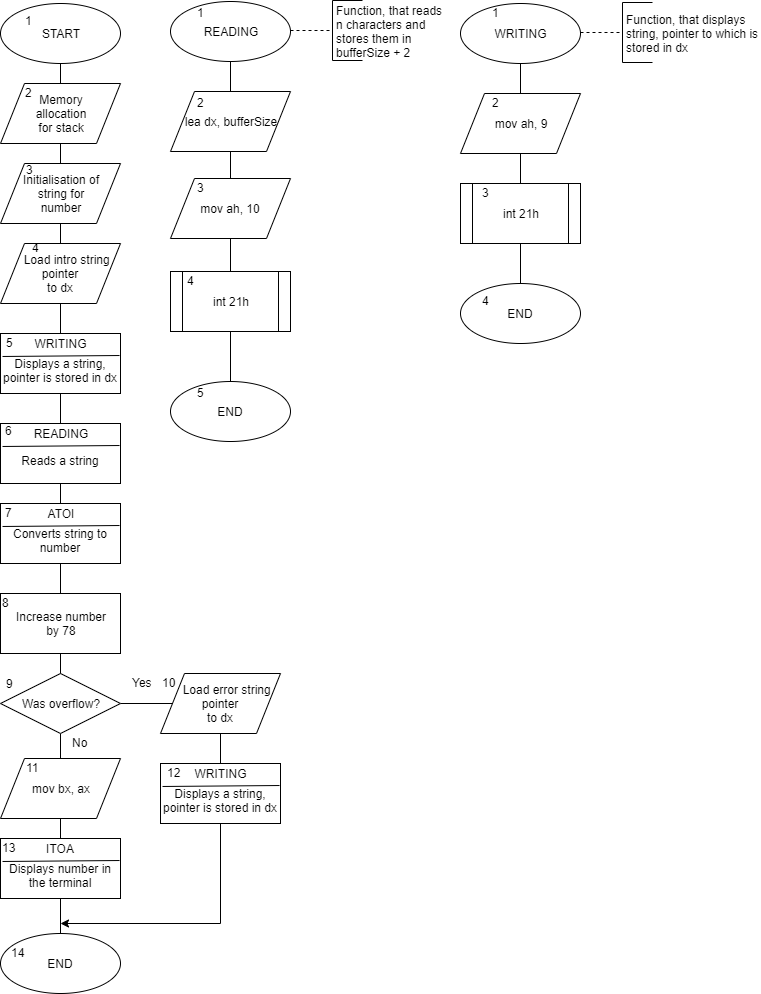
**Скріншоти:**

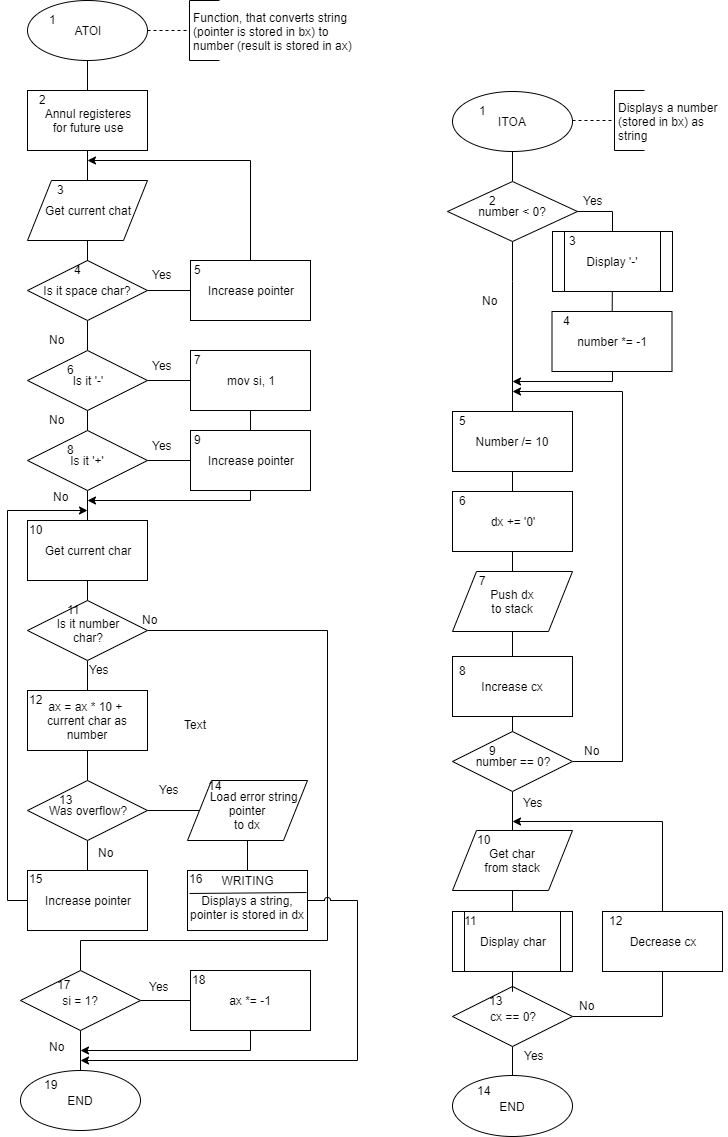
**E:\VM\SharedDir\Screenshots\lab02\Positive.PNG**

**E:\VM\SharedDir\Screenshots\lab02\Negative.PNG**

**E:\VM\SharedDir\Screenshots\lab02\Invalid.PNG**

**Блок-схема**:





Висновки:

1. Було засвоєно технологію використання переривань для зчитування строк, виведення строк та окремих символів
2. Було засвоєно використання цикла loop
3. Були написані окремі процедури, кожна з яких виконували свою роботу, а потім викликалися у коді. Ці процедури можуть бути використані у подальших комп’ютерних практикумах
4. Загалом, був засвоєний синтаксис асемблеру