

**Міністерство освіти і науки України**  
**Національний університет «Запорізька політехніка»**

кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №4

з дисципліни «Архітектура програмування та низькорівневе програмування»  
на тему: «Консольне введення/виведення числових даних. Операції  
арифметики, логіки та побітових зсувів. Умовні конструкції та цикли»

Виконала:

студентка групи КНТ-132

Мілена Кочева

Прийняв:

ст. викладач

О. І. Качан

### **Завдання до роботи:**

1. Розробити початковий код програми, яка повинна виконувати наступні функції:

- а) виведення повідомлення до консолі;
- б) зчитування з консолі текстового рядка;
- в) інтерпретація (парсинг) зчитанного текстового рядка як багаторозрядного числа визначеної на Ваш розсуд системи зчислення (2,10,16,36);
- г) варіативно можна увести не одно, а декілька чисел, а потім виконати будь-які на Ваш розсуд математичні операції над одним або декількома числами і отримати результуюче число;
- д) перетворити саме це результуюче число визначеної на Ваш розсуд (нової/іншої) системи зчислення (2,10,16,36) до текстового рядка;
- ж) виведення отриманого текстового рядка до консолі.

2. Дозволяється у коді реалізувати МЕНЮ, додавання до коду необхідних коментарів, відтворення покрокового алгоритму (у звіті), компіляція, лінування, запуск на виконання з демонстрацією роботи нормальних та виключних ситуацій (некоректне введення даних і програмне переповнення), самостійне та оригінальне авторське створення будь-яких дій над уведеними числами - або арифметичних (+, -, \*, /, возведення в ступінь), або логічних (AND, OR, XOR, NOT), або операцій побітових зсувів (<<, >>, циклічних). У звіті обов'язкова перевірка виконаних дій у ручному форматі, реалізація коду перевірки виключних ситуацій переповнення або виходу за межі діапазону вхідних чи вихідних числових даних.

3. Оформити звіт згідно ДСТУ 3008:2015 та завантажити поряд з власноруч створеним початковим кодом програми.

## Скріншоти:

```
C:\TASM\41aba.asm
.model small

sseg segment para stack 'stack' ;объявление стека
    db 256 dup(?)
sseg ends

dseg segment para public 'data' ; объявление сегмента данных
    crlf db 0Dh,0Ah,'$' ; переход на след строчку
    prmt db ' Input value','$' ; вывод текста "Input value"
    mesA db ' A:','$' ; вывод текста "A:"
    mesB db ' B:','$' ; вывод текста "B:"
    resM db 8 dup(' '), 'Result:','$' ; вывод текста "Result:"
    errF db ' Format Input Error','$' ; вывод текста в случае ошибки
    buff db 32 ; максимальный размер буферу
    blen db ? ; размер строчки после считывания
    binp db 32 dup(' '), '$' ; вміст буферу
    bout db 32 dup(' '), '$' ; вихідний рядок
dseg ends

cseg segment para public 'code' ; объявление сегмента кода
    assume cs:cseg,ss:sseg,es:nothing
begin proc far
    assume ds:dseg
    mov bx,dseg
    mov ds,bx

    lea dx,crlf ; показатель на переменную crlf (далее везде тоже показатели на разные переменные)
    call output ; вызов функции output

    lea dx,prmt
    call output

    lea dx,mesA
    call output

    lea dx,buff
    call input

    lea dx,crlf
    call output

    call to10num ; вызов функции to10num
    push ax

    lea dx,prmt
    call output

    lea dx,mesB
    call output
```

```

    lea dx, buff
    call input

    lea dx, crlf
    call output

    call to10num
    push ax

    pop bx    ; В у ВХ
    pop ax    ; А у АХ

; =====
; add bx, ax    ; додавання А + В (ax+bx)
; =====

    push ax    ; ax у stack

    lea dx, crlf    ; получаем показатель на переменную crlf
    call output    ; вызов функции output

    lea dx, resM    ; получаем показатель на переменную resM
    call output

    pop ax    ; вернуть АХ

    call to10str    ; вызов функции

    lea dx, bout    ; показатель на переменную bout
    add dx, di    ; прибавить dx к di
    call output    ; вызов функции output

    lea dx, crlf
    call output

    mov ah, 4Ch
    mov al, 00h    ; exit ot OS
    int 21h
begin endp

```

```

output proc near
    xor ax, ax    ; ax => 0
    mov ah, 09h
    int 21h
    ret
output endp

input proc near
    xor ax, ax
    mov ah, 0Ah
    int 21h
    ret
input endp

; перевод символа в десятичное число
to10num proc near
    xor ax, ax    ; АХ = 0
    xor bx, bx    ; ВХ = 0
    xor cx, cx    ; СХ = 0
    xor di, di    ; DI = 0 – номер байта в буфере
    mov cl, blen    ; CL = длина рядка
    mov si, cx    ; SI = последняя позиция в строке
    mov cx, 10    ; CX = 10, система вычисления
asc2bin:
    mov bl, byte ptr binp[di]    ; переназначения типа байту
    sub bl, '0'    ; число = код символу ASCII – 30h ('0')
    jb asc_err    ; если (код символа < код '0'),
    cmp bl, 9    ; если (число > 9), то
    ja asc_err    ; вызывает выход программы в проводник
    mul cx    ; иначе – умножить поточный результат на 10
    add ax, bx    ; добавить к поточному результату новый разряд числа
    inc di    ; перейти к след байту в буфере
    cmp di, si    ; если считано SI, последний символ в рядке, то выход
    jb asc2bin
    ret
asc_err:
    call errFormat
to10num endp

```

```

;перевод десятичного числа в рядок
to10str proc near
    xor dx,dx
    mov dl,buff
    dec dl
    mov di,dx          ; начальная позиция в рядку
    mov cx,10          ; CX = 10, система считывания
m1:
    cmp ax,cx          ; если (число < 10),
    jb m2              ; то закончить превращение
    xor dx,dx          ; DX=0 для выключения использования регистровой пары DX:AX
    div cx              ; беззнаковое деление AX и CX(10): результат у AX, остаток в DX
    add dx,'0'          ; превращение остатка до кода символу: 0->'0',1->'1'...9->'9'
    mov bout[di],dl     ; символ остатка всегда < 39h. Сохранил символ в рядку -а позиция DI
    dec di              ; замена позиции в рядку DI = DI - 1
    jmp m1              ; безусловный переход к M1
m2:
    add ax,'0'          ; превращение части к концу символу: 0->'0',1->'1'...9->'9'
    mov bout[di],al     ; символ части всегда < 39h, сохранил символ в рядку -а на позицию DI
    ret
to10str endp

errFormat proc near
    lea dx,errF         ; вызов показателя на символьный массив с текстом ошибки
    call output          ; вызов функции output
    mov ah,4Ch           ; выход в ОС
    mov al,01h
    int 21h
errFormat endp

cseg ends
end begin

```

C:\TASM>4LABA.EXE

Input value A:11  
Input value B:100

Result:111

C:\TASM>4LABA.EXE

Input value A:bukva  
Format Input Error  
C:\TASM>4laba.exe

Input value A:-10  
Format Input Error  
C:\TASM>