**Міністерство освіти і науки України**

**Національний університет «Запорізька Політехніка»**

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №4

з дисципліни «Архітектура Комп’ютера та Низькорівневе Програмування»

На тему «КОНСОЛЬНЕ ВВЕДЕННЯ/ВИВЕДЕННЯ ЧИСЛОВИХ ДАНИХ. ОПЕРАЦІЇ АРИФМЕТИКИ, ЛОГІКИ ТА ПОБІТОВИХ ЗСУВІВ. УМОВНІ КОНСТРУКЦІЇ ТА ЦИКЛИ»

Варіант №20

**Виконав:**

Студент групи КНТ-122 О. А. Онищенко

**Прийняли:**

Ст. Викладач О. І. Качан

Доцент А. Є. Казурова

2023

[Текст завдання 3](#_Toc135643331)

[Код програми 4](#_Toc135643332)

[Приклад роботи 6](#_Toc135643333)

Текст завдання

1. Розробити початковий код програми, яка повинна виконувати наступні функції:
   1. виведення повідомлення до консолі;
   2. зчитування з консолі текстового рядка;
   3. інтерпретація (парсинг) зчитанного текстового рядка як багаторозрядного числа визначенної на Ваш розсуд системи зчислення (2,10,16,36);
   4. 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;
   5. перетворити саме це результуюче число визначенної на Ваш розсуд (нової/іншої) системи зчислення (2,10,16,36) до текстового рядка;
   6. виведення отриманого текстового рядка до консолі.
2. Дозволяється у коді реалізувати МЕНЮ, додавання до коду необхідних коментарів, відтворення покрокового алгоритму (у звіті), компіляція, лінкування, 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 уведенними 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.

Код програми

; Sets up the memory model and stack size

.model small

.stack 100h

; Data segment

.data

    ; Introduction message

    intro db 0dh, 0ah, "Onyshchenko Oleh", 0dh, 0ah, "$"

    ; Name of the group

    group\_name db "     KHT-122", 0dh, 0ah, "$"

    ; Prompt for first number

    msg1 db 0dh, 0ah, "Enter the first number: $"

    ; Prompt for second number

    msg2 db 0dh, 0ah, "Enter the second number: $"

    ; Result message in decimal

    result\_dec db 0dh, 0ah, "Result in decimal: $"

    ; Result message in binary

    result\_bin db 0dh, 0ah, "Result in binary: $"

    ; Buffer for user input

    buffer db 9 dup ('$')

    ; End of line character

    endl db 0dh, 0ah, "$"

; Code segment

.code

; Entry point of the program

.startup

    ; Sets up the data segment

    mov ax, @data

    mov ds, ax

    ; Prints the introduction message

    mov ah, 09h

    lea dx, intro

    int 21h

    ; Prints the group name

    lea dx, group\_name

    int 21h

    ; Prompts the user for the first number

    mov ah, 09h

    lea dx, msg1

    int 21h

    ; Reads the user input for the first number

    mov ah, 01h

    int 21h

    sub al, '0'

    mov bl, al

    ; Prompts the user for the second number

    mov ah, 09h

    lea dx, msg2

    int 21h

    ; Reads the user input for the second number

    mov ah, 01h

    int 21h

    sub al, '0'

    mov cl, al

    ; Adds the two numbers

    add bl, cl

    ; Prints the result in decimal

    mov ah, 09h

    lea dx, endl

    int 21h

    lea dx, result\_dec

    int 21h

    ; Print the value of BL in ASCII

    mov dl, bl

    add dl, '0'

    mov ah, 02h

    int 21h

    ; Convert the value of BL from decimal to binary and store it in the buffer

    mov ch, 8         ; set the loop counter

    mov si, offset buffer  ; point to the buffer

    mov cl, bl        ; store the value of BL in CL

    convert\_loop:

        mov al, cl            ; move the value of CL into AL

        and al, 10000000b     ; mask the most significant bit

        shr al, 7             ; shift the bit to the least significant bit

        or al, 00110000b      ; add the ASCII code for '0' or '1'

        mov [si], al          ; store the result in the buffer

        inc si               ; increment the buffer pointer

        shl cl, 1             ; shift the bits in CL to the left

        dec ch               ; decrement the loop counter

        jnz convert\_loop      ; repeat until all 8 bits have been processed

    ; Print the binary representation of BL stored in the buffer

    mov ah, 09h

    lea dx, result\_bin

    int 21h

    ; Print the contents of the buffer

    mov dx, offset buffer

    mov ah, 09h

    int 21h

    lea dx, endl

    int 21h

    ; Exit the program

    mov ah, 4ch

    int 21h

; End of program label

.exit

end

Приклад роботи



