

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

Звіти до комп'ютерних практикумів дисципліни

«Системне програмне забезпечення»

Прийняв
доцент кафедри ІІІ
Лісовиченко О.І.
“26” травня 2023 р.

Виконав
Студент групи ІІІ-11
Прищеп В.С.

Київ-2023

Комп'ютерний практикум №2

Тема: засоби обміну даними.

Завдання:

1. Написати програму з використанням 2-х процедур:

1) Процедура введення і перетворення цілого числа. Після цього треба виконати математичну дію над числом (номер завдання вибирати за останніми двома числами номеру в заліковій книжці - Таблиця 2.1).

2) Процедура переведення отриманого результату в рядок та виведення його на екран.

2. Програма повинна мати захист від некоректного введення вхідних даних (символи, переповнення, ділення на 0 і т.і.).

№ в зал. книжці ариф. дія	№ в зал. книжці ариф. дія	№ в зал. книжці ариф. дія	№ в зал. книжці ариф. дія
1. +67	2. +15	3. +5	4. -78
5. -32	6. +78	7. *9	8. -23
9. *4	10. +23	11. -88	12. -99
13. +124.	14. *7	15. *3	16. *7
17. -8	18. +6	19. -34	20. -23
21. 6. *3	22. *2	23. +47	24. -96
25. *23	26. -78	27. +78	28. *5
29. +33	30. -44	31. +55	32. -99
33. *15	34. -77	35. -77	36. +90

Варіант №21

Текст програми:

```
STSEG SEGMENT PARA STACK "STACK"  
DB 64 DUP("STACK")  
STSEG ENDS
```

```
DSEG SEGMENT PARA PUBLIC "DATA"  
inarr db 7,?,7 dup (" $")  
input_tip db 13, 10, "x in [-10922; 10922] => $"  
is_negative db 0  
num dw 0  
digit dw 0  
is_error db 0  
error_message_1 db "Invalid symbol(s)!$"  
error_message_2 db "Number out of diapason!$"  
DSEG ENDS
```

```
CSEG SEGMENT PARA PUBLIC "CODE"  
ASSUME CS:CSEG, DS:DSEG, SS:STSEG
```

```
main proc
```

```
;ds initialisation  
mov ax, dseg
```

```

    mov ds, ax

;showing a tip
    lea dx, input_tip
    mov ah, 9
    int 21h

;input of number
    call input_digit
    call str_to_int
    cmp is_error, 1
    je raise_error_1
    cmp is_error, 2
    je raise_error_2

;3*num
    mov ax, num
    mov bx, 3
    imul bx
    jo raise_error_2
    mov num, ax

;printing of the result
    call result_print
    jmp end_program

;invalid symbol(s) error
raise_error_1:
    LEA dx, error_message_1
    MOV ah,9
    INT 21h
    jmp end_program

;wrong number error
raise_error_2:
    LEA dx, error_message_2
    MOV ah,9
    INT 21h

;program finishing
end_program:
    mov AH, 4CH
    int 21H
    ret
main endp

```

input_digit proc

;input a string

lea dx, inarr

mov ah, 10

int 21h

;print a new line after the input

mov al,10

int 29h

mov al,13

int 29h

ret

input_digit endp

str_to_int proc

;load the address of first element of the array

mov si, offset inarr + 2

mov cx, 0

;check if inarr is empty

mov ax, 0

mov al, inarr + 1

cmp al, 0

je error_1

;converting of the array into number

convert_loop:

;check if it is the end of inarr

mov ax, 0

mov al, inarr + 1

cmp al, cl

je last

;set element of inarr to al

mov al, [si]

;check if the character is between "0" and "9"

cmp al, '0'

jl check_minus

cmp al, '9'

jg error_1

inc cx

inc si

```
;jump for converting character to digit
    jmp convert_to_digit
```

```
;check character for "-"
    check_minus:
        cmp al, '-'
        jne error_1
```

```
;check if it is the first element in inarr
    cmp cx, 0
    jne error_1
    mov is_negative, 1
    inc cx
    inc si
    jmp convert_loop
```

```
;converting character to digit
    convert_to_digit:
        sub al, '0'
        mov digit, ax
        mov bx, 10
        mov ax, num
        mul bx
        jc error_2
        js error_2
        mov num, ax
        mov ax, digit
        add num, ax
        jc error_2
        js error_2
        jmp convert_loop
```

```
;Used wrong symbol(s)
    error_1:
        mov is_error, 1
        jmp finish
```

```
;Inputed number out of range
    error_2:
        mov is_error, 2
        jmp finish
```

```
;check if the number is negative
    last:
        cmp is_negative, 1
```

```

    jne finish
    neg num

finish:
    ret
str_to_int endp

result_print proc

;check if result is negative
    mov bx, num
    or bx, bx
    jns m1

;if is negative, print '-' and make it positive
    mov al, '-'
    int 29h
    neg bx

;prepare to split number into digits
m1:
    mov ax, bx
    xor cx, cx
    mov bx, 10

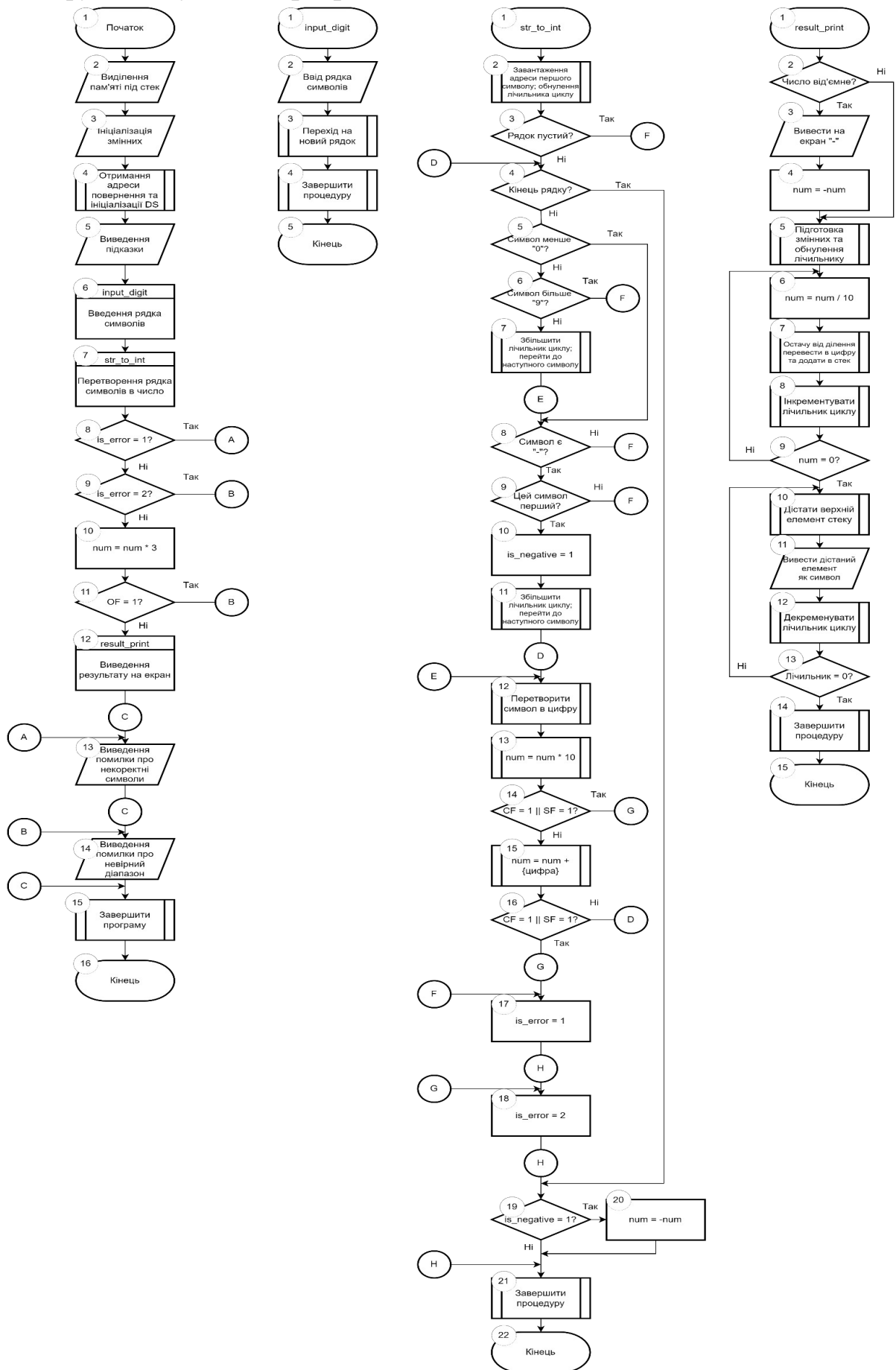
;split it into digits and write it down
m2:
    xor dx, dx
    div bx
    add dl, '0'
    push dx
    inc cx
    test ax, ax
    jnz m2

;print these digits in correct order
m3:
    pop ax
    int 29h
    loop m3
    ret
result_print endp

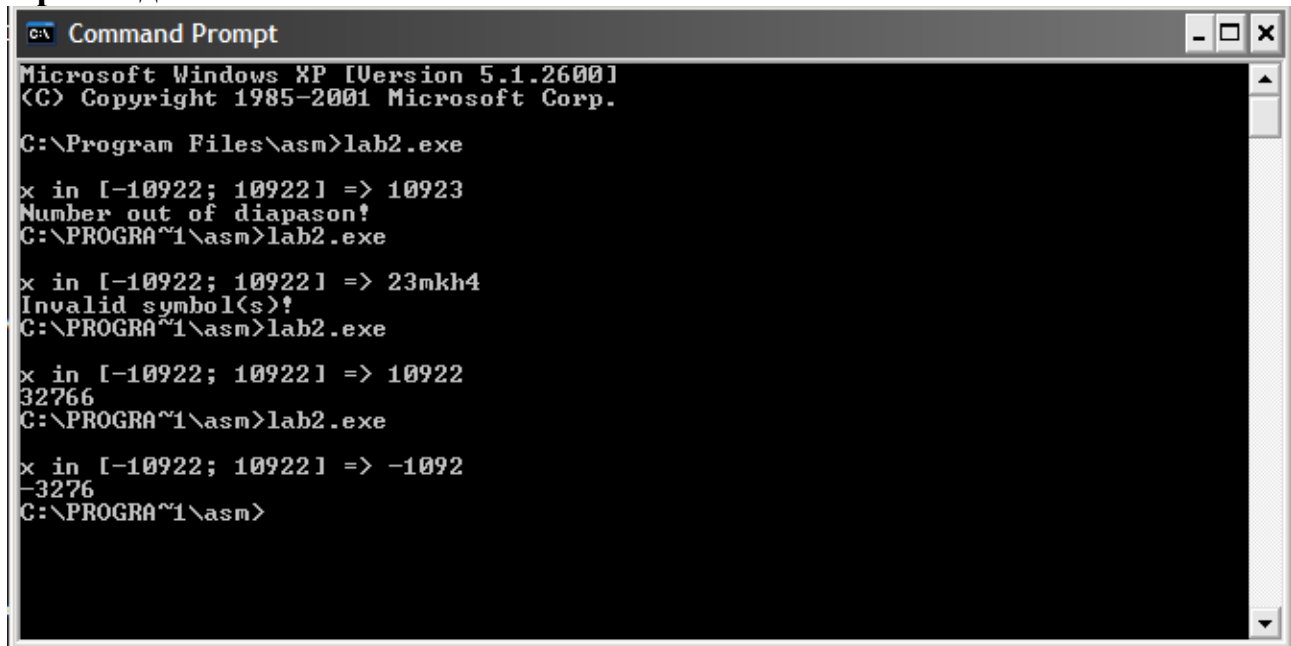
CSEG ENDS
END MAIN

```

Схема функціонування програми:



Приклад виконання:



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\asm>lab2.exe
x in [-10922; 10922] => 10923
Number out of diapason!
C:\PROGRA~1\asm>lab2.exe
x in [-10922; 10922] => 23mkh4
Invalid symbol(s)!
C:\PROGRA~1\asm>lab2.exe
x in [-10922; 10922] => 10922
32766
C:\PROGRA~1\asm>lab2.exe
x in [-10922; 10922] => -1092
-3276
C:\PROGRA~1\asm>
```

У першому виклику введене значення було поза дозволеного діапазону, в другому — було введене не число, а в третьому та четвертому — були введені правильні числа і програма вивела не повідомлення про помилку, а результат.

Висновок:

1. Написана програма з використанням 3-х процедур:

- 1) Процедура введення рядку.
- 2) Процедура переведення рядка символів в число.
- 3) Процедура переведення отриманого результату в рядок та виведення його на екран.

Дія над числом виконується між 2 та 3 процедурою згідно з варіантом.

2. Програма має захист від некоректного введення вхідних даних (символи, переповнення). Результати захисту продемонстровані в звіті.