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 Виконав**

**доцент кафедри ІПІ Студент групи ІП-11**

**Лісовиченко О.І. Головня О.Р.**

**“12” травня 2023 р.**

Київ – 2023

**Комп’ютерний практикум №2**

**Тема**: Засоби обміну даними

**Завдання**:

1. Написати програму з використанням 2-х процедур:

Процедура введення і перетворення цілого числа. Після цього треба виконати математичну дію над числом.

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

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

Варіант 6 (+78)

sseg segment para stack 'stack'

db 64 dup ( 'STACK' )

sseg ends

dseg segment para public 'data'

is\_ok dw 1

num dw 0

temp dw 0

is\_negative dw 0

prompt db 0dh, 0ah,'enter number in range of -32767 to 32689: $'

result db 0dh, 0ah, 'result: $'

input\_error\_message db 0dh, 0ah, 'the number you entered is too large$'

addition\_error db 0dh, 0ah, 'the result of addition is more than 32767$'

wrong\_character\_message db 0dh, 0ah, 'wrong character$'

empty\_input\_message db 0dh, 0ah, 'empty input$'

dseg ends

cseg segment para public 'code'

main proc

assume cs: cseg, ds: dseg, ss: sseg

push ds

xor ax, ax

push ax

mov ax, dseg

mov ds, ax

call read\_digit

call add\_78

cmp is\_ok,1

jne not\_equal

jmp end\_if

not\_equal:

jmp finally

end\_if:

call print\_digit

finally:

mov ah, 4ch

int 21h

main endp

add\_78 proc

add num, 78

jo add\_error ; x+78>32768

ret

add\_error:

mov is\_ok, 0

lea dx, addition\_error

mov ah, 09h

int 21h

ret

add\_78 endp

read\_digit proc

lea dx, prompt

mov ah, 09h

int 21h

mov bx, 10

mov cx, 5

read:

xor ax, ax

mov ah, 01h

int 21h

cmp al, 13 ;if enter -> stop

je stop

cmp al, 48 ;if n -> check\_s

jl check\_sign

cmp al, 57

ja wrong\_character ;if !n -> stop

sub al, '0' ;get n value

sub ah, ah ;zeoring the high byte of AX reg

mov temp, ax ;save n

mov ax, num

imul bx

jo input\_error

add ax, temp

jo input\_error

mov num, ax

loop read

cmp is\_negative, 1

je make\_negative

ret

stop:

cmp cx, 5

je empty\_input

mov cx, 0

cmp is\_negative, 1

je make\_negative

ret

make\_negative:

neg num

ret

check\_sign:

cmp al, '-'

jne wrong\_character

cmp cx, 5

jne wrong\_character

mov is\_negative, 1

jmp read

input\_error:

mov is\_ok, 0

lea dx, input\_error\_message

mov ah, 09h

int 21h

jmp fin

wrong\_character:

mov is\_ok, 0

lea dx, wrong\_character\_message

mov ah, 09h

int 21h

jmp fin

empty\_input:

mov is\_ok, 0

lea dx, empty\_input\_message

mov ah, 09h

int 21h

fin:ret

read\_digit endp

print\_digit proc

lea dx, result

mov ah, 09h

int 21h

mov bx, num

or bx, bx ;the sign bit in the bx reg SET

jns m1

mov al, '-'

int 29h ;display -

neg bx

m1:

mov ax, bx

xor cx, cx ;cx used as a counter of numbers dec

mov bx, 10

m2: ;divide by 10 and add 0

xor dx, dx ; repeat until the int part will not = 0

div bx

add dl, '0'

push dx

inc cx

test ax, ax ;=0

jnz m2

m3: ; begin printing from the most significant digit

pop ax

int 29h

loop m3

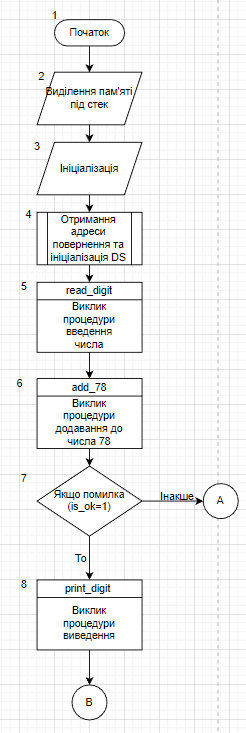
ret

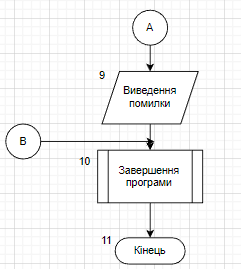
print\_digit endp

cseg ends

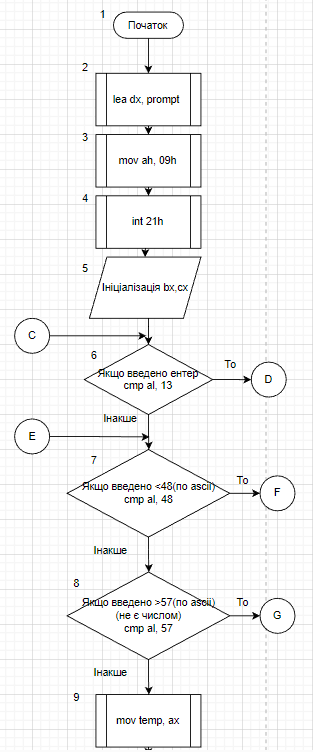
end main

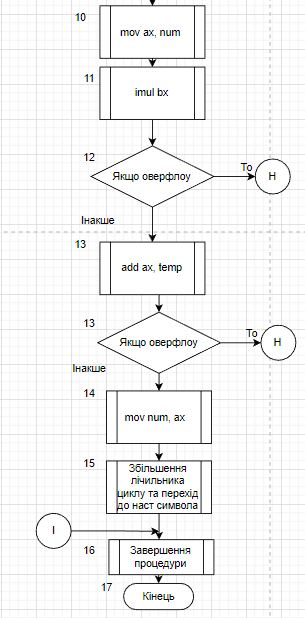
Блок-схема програми:

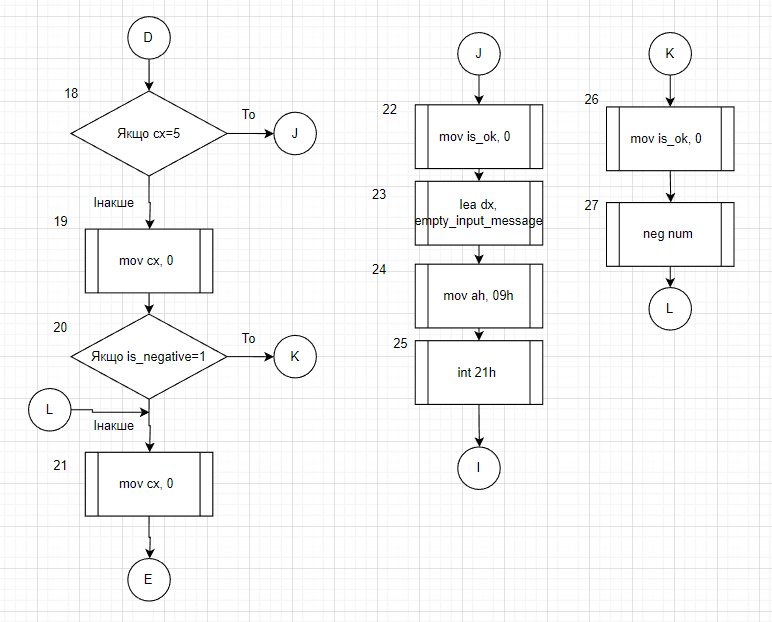
****

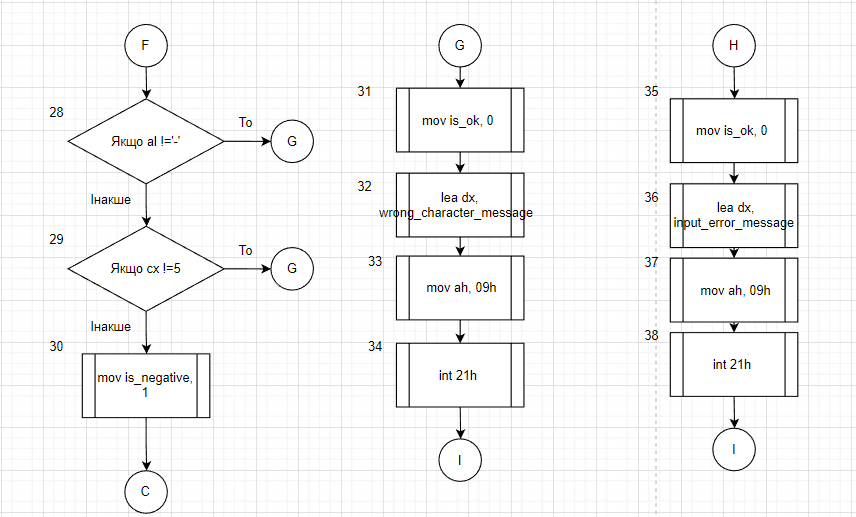


Блок-схема процедури read\_digit:

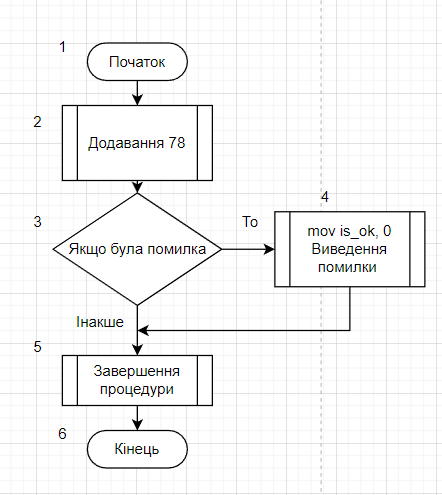


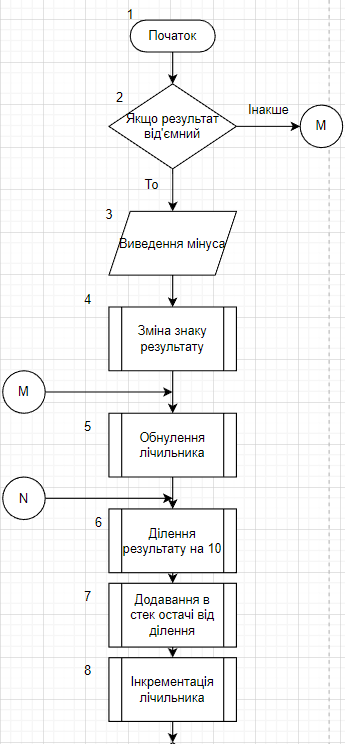


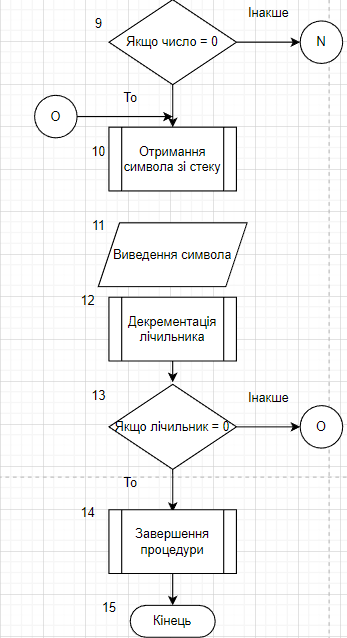




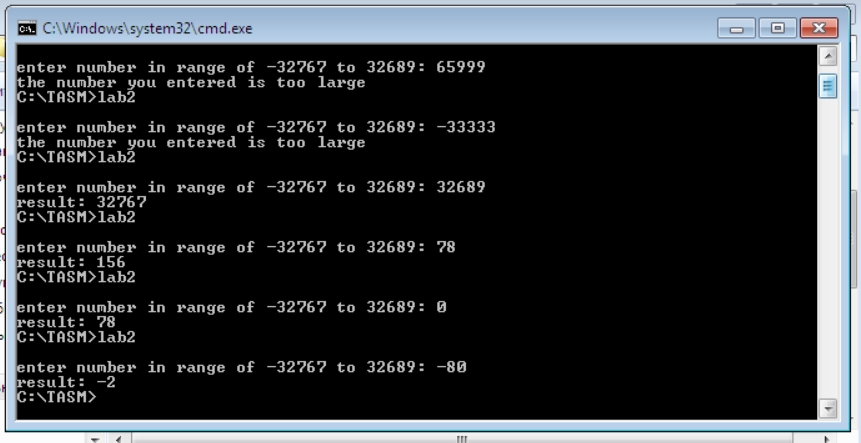
Блок-схема процедури add\_78



Блок-схема процедури print\_digit:  




**Тестування програми:**

Для додавання числа 78 діапазон можливих вхідних значень має бути [-32767; 32689], щоб діапазон результатів був [-32767; 32767].  
  
  
Висновок: В текстовому редакторі написав програму за своїм варіантом завдання в файлі .asm. Програма додає число 78 від введеного користувачем числа та виводить на екран результат. Також, побудував блок-схему за кодом програми та протестував програму на коректність результатів обчислень та захист від некоректного введення вхідних даних. Таким чином, програма коректно виконує поставлену задачу