

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з комп'ютерного практикуму № 2 з дисципліни

«Системне програмне забезпечення»

«Засоби обміну даними»

Виконав:
студент групи ІП-11
Лесів В. І.

Перевірив:
Лісовиченко О. І.
«10» квітня 2023 р.

Київ 2023

Комп'ютерний практикум 2

Засоби обміну даними

Постановка завдання.

1. Написати програму з використанням 2-х процедур:
 - а. Процедура введення і перетворення цілого числа. Після цього треба виконати математичну дію над числом – множення на 7.
 - б. 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.).

Хід роботи.**Текст програми.**

```
stseg segment stack "stack"
```

```
db 256 dup ( "?" )
```

```
stseg ends
```

```
dseg segment public "data"
```

```
buff db 7, 8 dup("$")
```

```
invalid_input_mes db 13, 10, "Invalid input$"
```

```
enter_number_mes db 13, 10, "Enter number from [-4681; 4681]: $"
```

```
result_mes db 13, 10, "Result of multiplying by 7: $"
```

```
dseg ends
```

```
cseg segment public "code"
```

```
assume ds:dseg, cs:cseg, ss:stseg
```

```
start:
```

```
mov ax, dseg
```

```
mov ds, ax
```

```
program:
```

```
call read
```

```
mov bx, dx
```

```
lea si, [bx]+2
```

mov dl, [si]

call transform

mov dx, 7

imul dx

call output_result

end_program:

mov ah, 4Ch

int 21h

transform proc

push bp

mov bp, sp

mov ax, 10

push ax

mov cl, [bx]+1

lea si, [bx]+2

mov dl, [si] ;first char

```
cmp dl, '0' ;
```

```
je error
```

```
cmp dl, '-' ;
```

```
je negative_sign
```

```
mov bx, 0
```

```
push bx
```

```
jmp transform_number
```

```
negative_sign:
```

```
mov bx, 1
```

```
push bx
```

```
inc si
```

```
sub cx, 1
```

```
cmp si, 1
```

```
je error
```

```
transform_number:
```

```
xor ax, ax
```

```
xor dx, dx
```

```
loop_start:
```

mov dl, [si]

sub dl, '0'

jb error

cmp dl, 9

ja error

push dx

mov dx, [bp]-2

mul dx

pop dx

jc error

add ax, dx

js error

cmp ax, 4681

jg error

inc si

loop loop_start

mov dx, [bp]-4

cmp dx, 0

je end_procedure

neg ax

cmp ax, -4681

jle error

end_procedure:

mov sp, bp

pop bp

ret

transform endp

error:

mov ah, 9h

lea dx, invalid_input_mes

int 21h

jmp program

read proc

push bp

mov bp, sp

```
mov ah, 9h  
  
lea dx, enter_number_mes  
  
int 21h  
  
lea dx, buff  
  
mov ah, 0Ah  
  
int 21h  
  
pop bp  
  
ret  
  
read endp
```

```
output_result proc  
  
push bp  
  
mov bp, sp  
  
push ax  
  
mov ah, 9h  
  
lea dx, result_mes  
  
int 21h  
  
pop ax  
  
  
mov bx, ax  
  
or bx, bx  
  
jns m1  
  
mov al, '-'  
  
int 29h
```



```
neg bx

m1:

mov ax, bx

xor cx, cx

mov bx, 10

m2:

xor dx, dx

div bx

add dl, '0'

push dx

inc cx

test ax, ax

jnz m2

m3:

pop ax

int 29h

loop m3

pop bp

ret

output_result endp

cseg ends

end start
```

Схема функціонування програми.

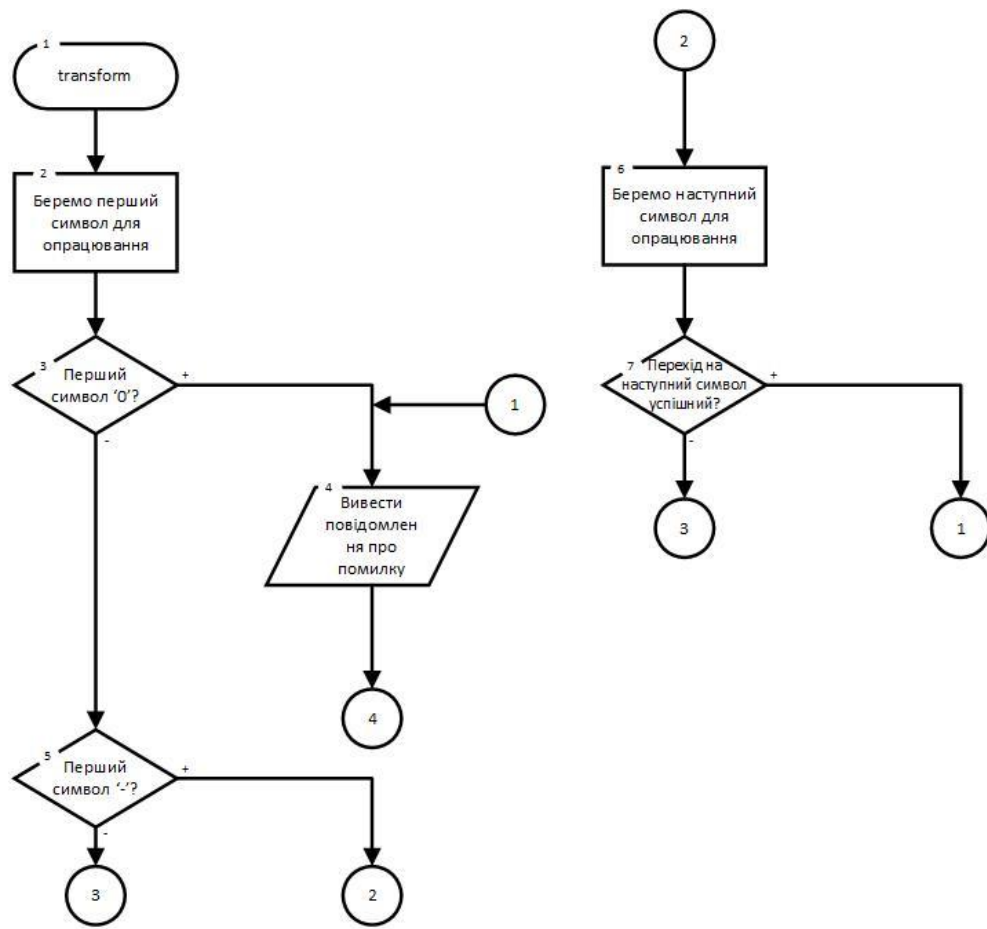


Рисунок 1. Схема процедури transform (початок)

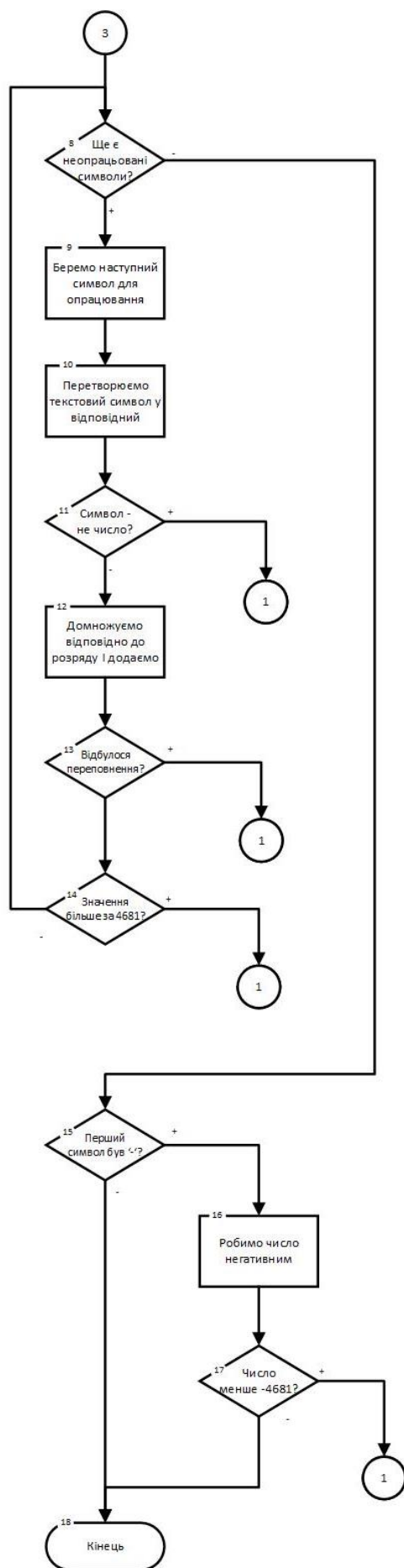


Рисунок 2. Схема процедури transform (кінець)



Рисунок 3. Схема процедури read

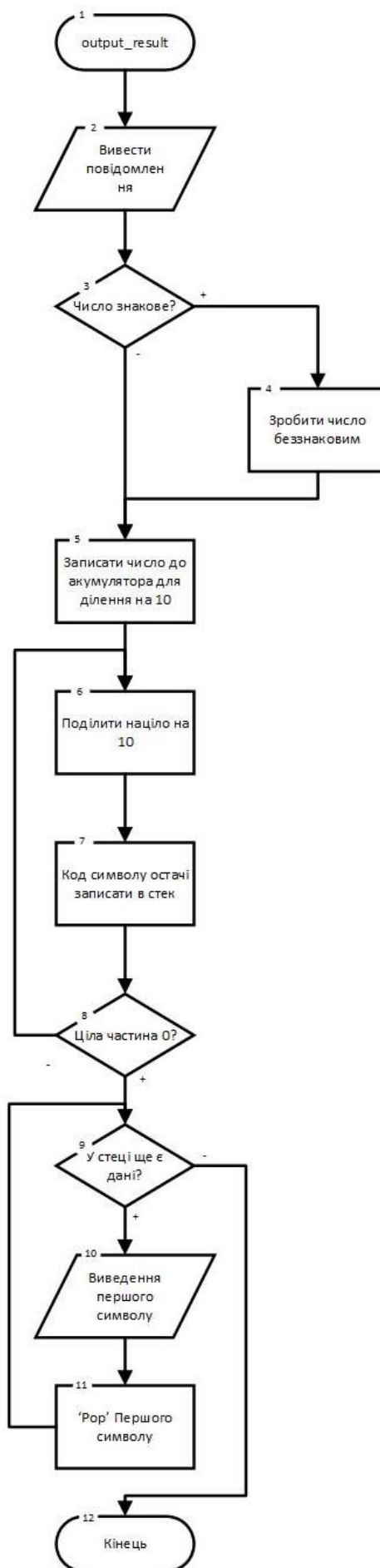
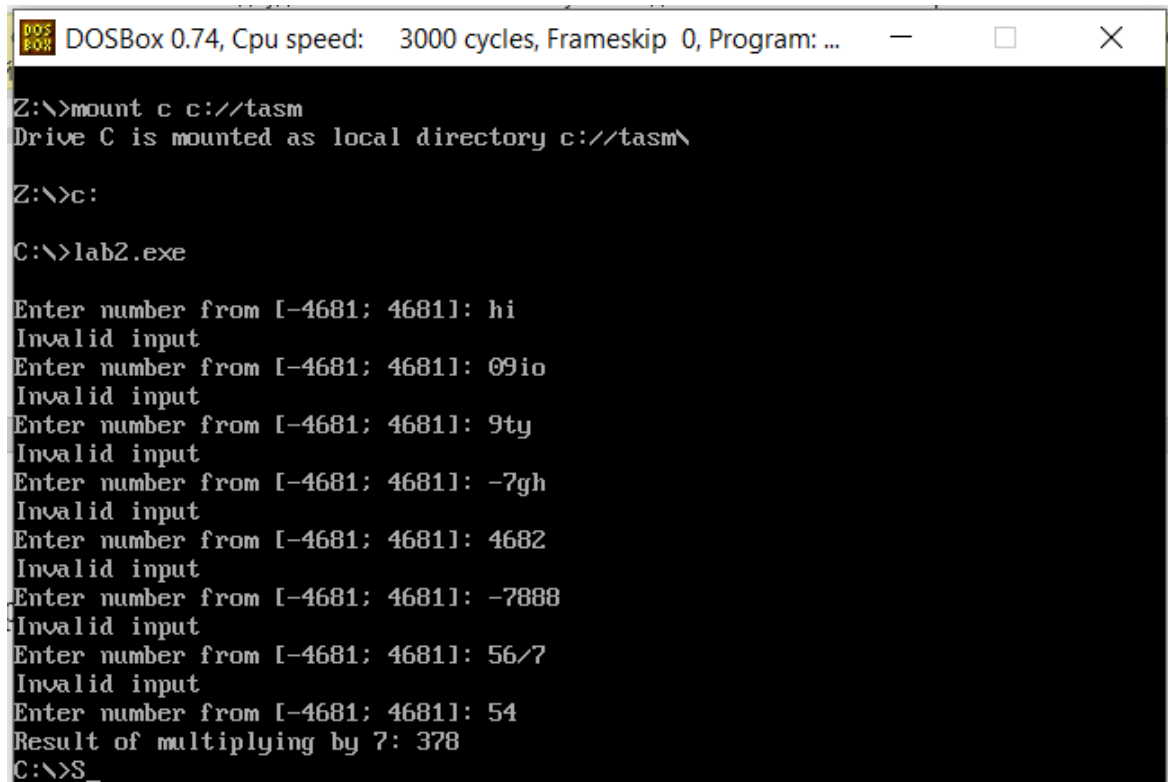
Рисунок 4. Схема процедури `output_result`



Рисунок 5. Схема основної частини програми.

Приклад виконання програми.



```

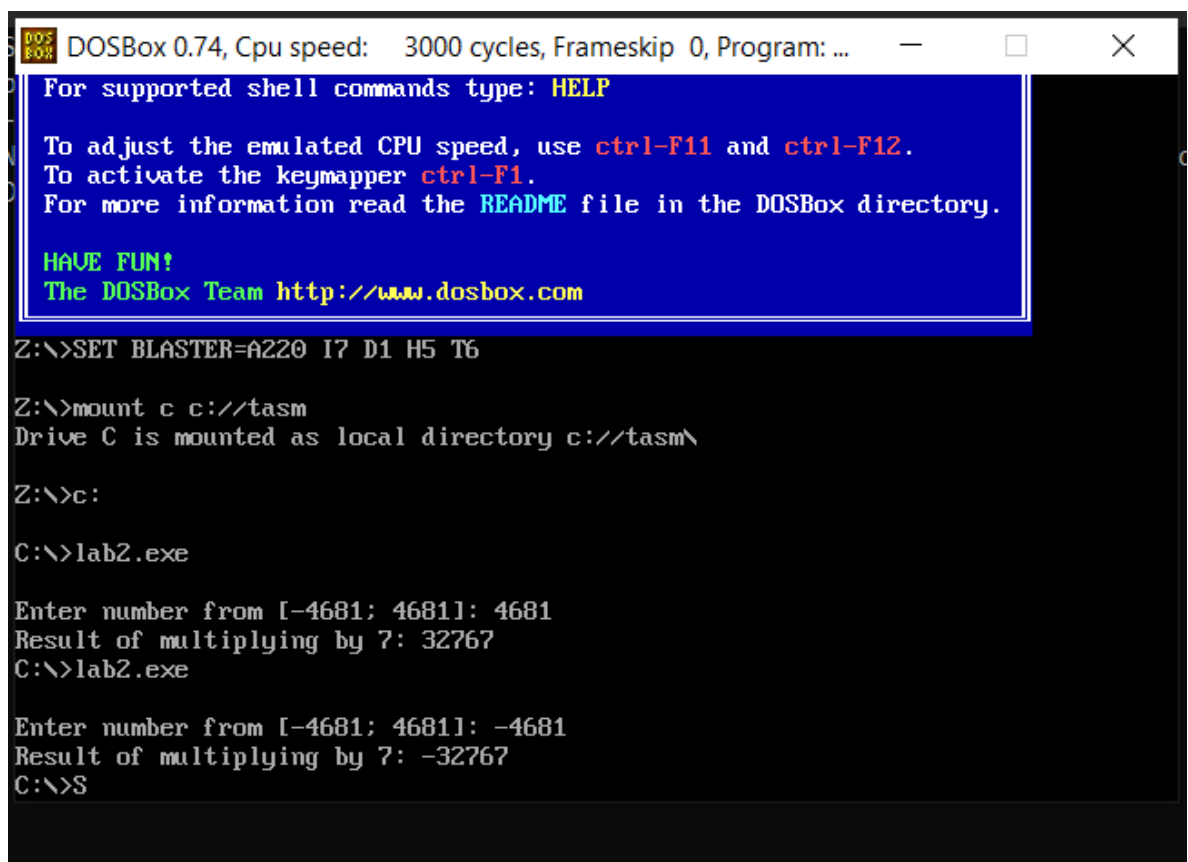
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
Z:\>mount c c://tasm
Drive C is mounted as local directory c://tasm\

Z:\>c:

C:\>lab2.exe

Enter number from [-4681; 4681]: hi
Invalid input
Enter number from [-4681; 4681]: 09io
Invalid input
Enter number from [-4681; 4681]: 9ty
Invalid input
Enter number from [-4681; 4681]: -7gh
Invalid input
Enter number from [-4681; 4681]: 4682
Invalid input
Enter number from [-4681; 4681]: -7888
Invalid input
Enter number from [-4681; 4681]: 56/7
Invalid input
Enter number from [-4681; 4681]: 54
Result of multiplying by 7: 378
C:\>S_
  
```

Рисунок 6. Приклад виконання програми з некоректними значеннями.



```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: ...
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c://tasm
Drive C is mounted as local directory c://tasm\

Z:\>c:

C:\>lab2.exe

Enter number from [-4681; 4681]: 4681
Result of multiplying by 7: 32767
C:\>lab2.exe

Enter number from [-4681; 4681]: -4681
Result of multiplying by 7: -32767
C:\>S
  
```

Рисунок 7. Приклад виконання програми з граничними значеннями.

Програма повідомляє про некоректне введення, якщо введено символи або ж значення, що перевищує задані межі. Якщо значення некоректне, програма повертається у початковий стан і очікує введення коректного значення. Після введення коректного значення програма виводить повідомлення про дію й результат виконання.

Висновок.

Отже, у даній роботі я ознайомився із засобами обміну даними.

Було написано програму з використанням процедур:

1. Процедури введення і перетворення цілого числа. Після цього виконано математичну дію над числом – помножити на 7.
2. Процедура переведення отриманого результату в рядок та виведення його на екран.

Програма має захист від некоректного введення вхідних даних (символи, переповнення). Програму було скомпільовано, налагоджено та виконано. У результаті виконання програми отримується коректний й очікуваний результат.