

Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки
Кафедра технічної кібернетики

*Звіти до комп'ютерних практикумів з модуля
«Системне програмування»*

Прийняв
доцент кафедри ТК
Лісовиченко О.І.
“...”2018 р.

Виконав
Студент групи ІТ-61
Лесогорський К.С.

Комп'ютерний практикум №3

Тема: програмування розгалужених алгоритмів.

Завдання:

1. Написати програму, яка буде обчислювати значення функції. Номер завдання вибирати за останніми двома числами номеру в заліковій книжці.

$$13. \quad Z = \begin{cases} (40x^2 - 23) / x & \text{якщо } 0 < x \leq 7 \\ 38x^3 + 5 & \text{якщо } x \leq 0 \\ 126 / x & \text{якщо } x > 7 \end{cases}$$

Текст програми:

```
%define check_overflow jo report_overflow
%define stdout 0x1
%define sys_write 0x4
%define stdin 0x2
%define sys_read 0x3
SECTION .data
    overflow_mes db 'overflow occurred during program runtime. Please, invent
exceptions to handle this properly',10,13
    len equ $ - overflow_mes
    input_mes db 'Please, enter a signed byte(or else you will break
something)',10,13
    leni equ $ - input_mes

SECTION .text

EXTERN read_num, print_num
GLOBAL _start

_start:
    mov eax, sys_write
    mov ebx, stdout
    mov ecx, input_mes
    mov edx, leni
    int 80h
    call read_num
    push ax
    call cmp_num
    add esp, 4
    push ax
```

```
call print_num
mov ax,0x1
int 0x80
```

cmp_num: ;expects short(16bit, signed)

```
push ebp
mov ebp,esp
mov dx,[ebp + 8]
cmp dx, 0
jle lte_zero
cmp dx,7
jle b_zero_seven
jmp gt_seven
```

lte_zero:

```
xor dx,dx; no imul today!
mov ax,[ebp + 8]
neg ax
mov bx,[ebp + 8]
neg bx
mul bx
check_overflow
mul bx
check_overflow; x^3
mov bx, 38
mul bx
neg ax
check_overflow
add ax,5
check_overflow
jmp finally
```

gt_seven:

```
xor dx,dx
mov ax, 126
mov bx,[ebp + 8]
div bx
xor ah,ah
jmp finally
```

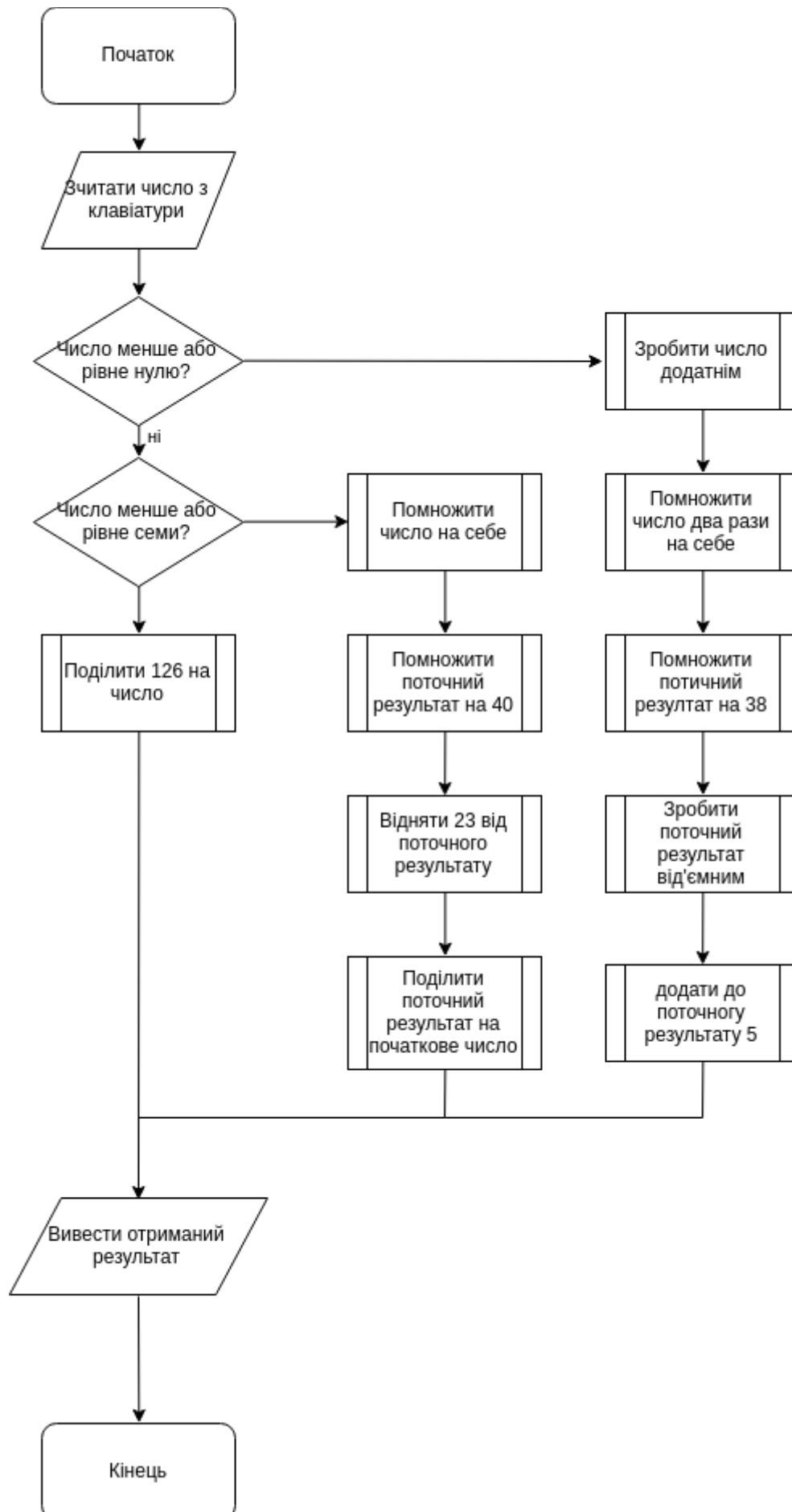
b_zero_seven:

```
xor dx,dx
mov ax,[ebp + 8]
mul ax
check_overflow
mov bx, 40
```

```
    mul bx
    check_overflow
    sub ax, 23
    mov bx,[ebp+8]
    div bx
    xor ah,ah
    jmp finally
finally:
    mov esp,ebp
    pop ebp
    ret

report_overflow:
    mov eax, sys_write
    mov ebx, stdout
    mov ecx, overflow_mes
    mov edx, len
    int 80h
    mov eax, 0x1
    int 80h
```

Схема функціонування програми:



Значення у секціях та регістрах:

Значення регістру ax до виконання обчислення:

```
(gdb) i r  
eax          0xa      10
```

Значення регістру ax після виконання обчислення:

```
(gdb) i r  
eax          0xc      12
```

Висновок:

1. В даному комп'ютерному практикумі було написано програму, яка включає процедуру для введення та перетворення з рядка в число цілого числа та обчислення математичної функції з використанням розгалуження.