

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №1
з дисципліни
«Алгоритми і структури даних»

Виконав:

Студент групи ІМ-44
Мундурс Нікіта Юрійович
номер у списку групи: 16

Перевірила:

Молчанова А. А.

Київ 2024

Завдання:

Задано дійсне число x . Визначити значення заданої за варіантом кусочно безперервної функції $y(x)$, якщо воно існує, або вивести на екран повідомлення про неіснування функції для заданого x .

Розв'язати задачу двома способами (створити дві блок-схеми алгоритмів та написати дві програми):

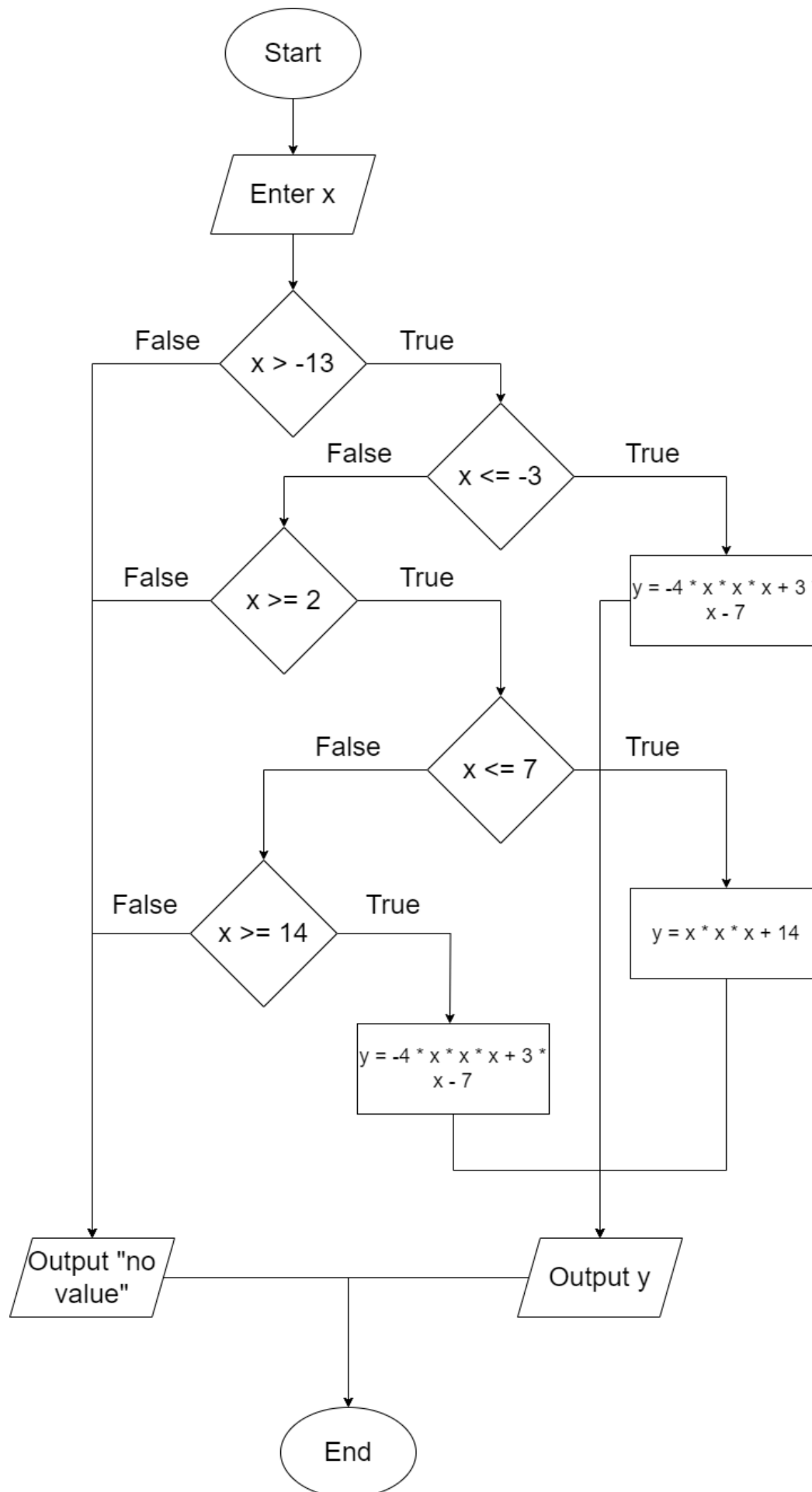
1) у програмі дозволяється використовувати тільки одиничні операції порівняння (\leq , $>$, \geq), і не дозволяється використовувати булеві (логічні) операції ($!$, $\&\&$, $||$ тощо);

2) у програмі необхідно обов'язково використати булеві (логічні) операції ($!$, $\&\&$, $||$ тощо); використання булевих операцій не повинно бути надлишковим.

Варіант 16:

$$y = \begin{cases} x^3 + 14, & x \in [2, 7] \\ -4x^3 + 3x - 7, & x \in (-13, -3] \cup (14, +\infty) \end{cases}$$

Діаграма алгоритму 1



Текст програми 1

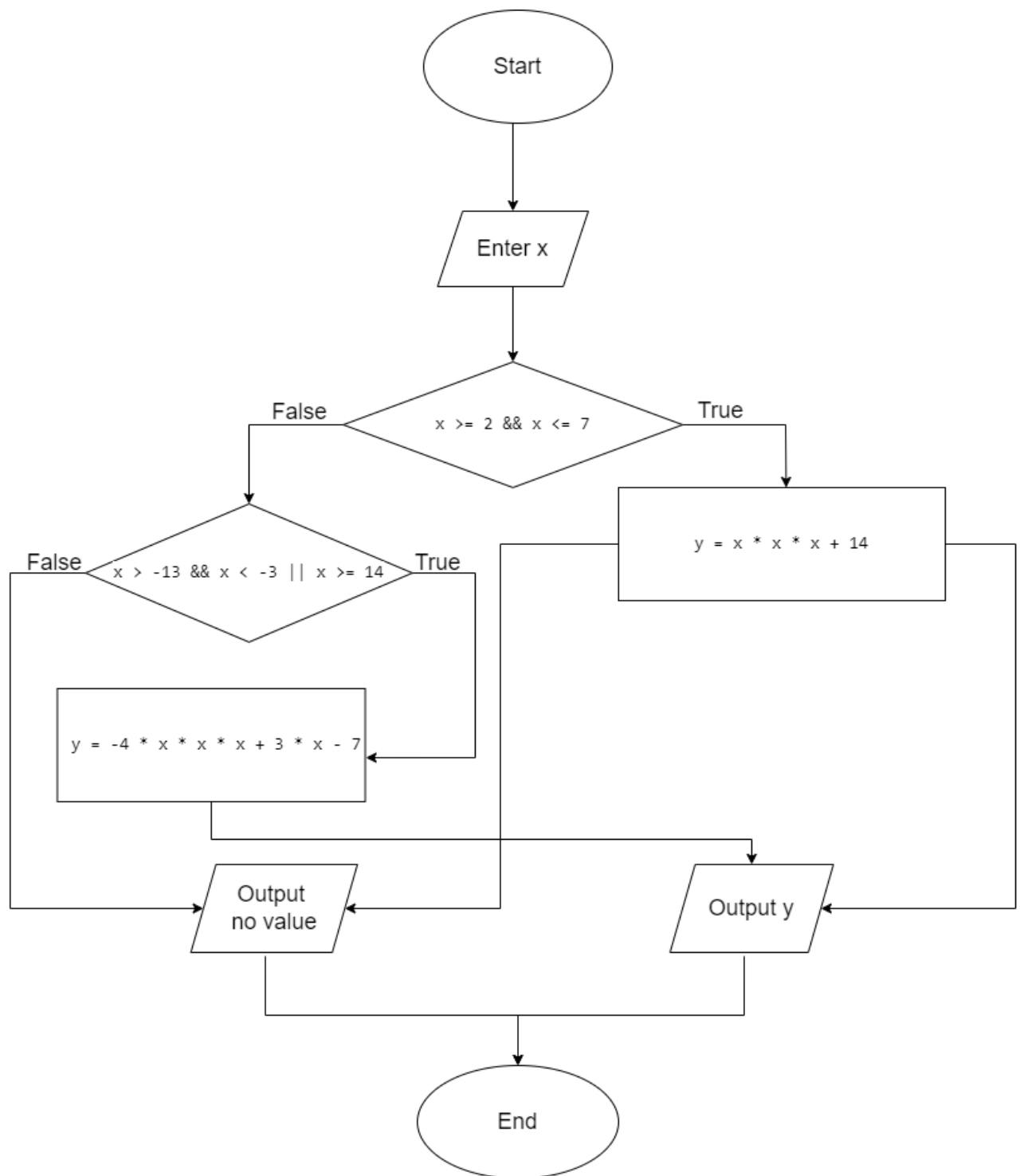
```
#include <stdio.h>

int main(int argc, char const *argv[]) {
    float x, y = 0;
    printf("Enter x: ");
    scanf("%f", &x);
    if (x > -13) {
        if (x <= -3) y = -4 * x * x * x + 3 * x - 7;
        else if (x >= 2) {
            if (x <= 7) y = x * x * x + 14;
            else if (x >= 14) y = -4 * x * x * x + 3 * x - 7;
        }
        printf("f(%.2f) = %.2f\n", x, y);
    }
    else printf("no value\n");
    return 0;
}
```

Результати тестування програми

```
PS D:\programming\code\lab1> gcc -o main.exe main1.c
PS D:\programming\code\lab1> .\main.exe
Enter x: -5
f(-5.00) = 478.00
no value
PS D:\programming\code\lab1> .\main.exe
Enter x: 1
no value
PS D:\programming\code\lab1> .\main.exe
Enter x: 4
f(4.00) = 78.00
no value
PS D:\programming\code\lab1> .\main.exe
Enter x: 10
no value
PS D:\programming\code\lab1> .\main.exe
Enter x: 20
f(20.00) = -31947.00
no value
PS D:\programming\code\lab1> █
```

Діаграма алгоритму 2



Текст програми 2

```
#include <stdio.h>

int main(int argc, char const *argv[]) {
    float x, y = 0;
    printf("Enter x: ");
    scanf("%f", &x);
    if (x >= 2 && x <= 7) {
        y = x * x * x + 14;
        printf("f(%.2f) = %.2f\n", x, y);
    }
    else if (x > -13 && x < -3 || x >= 14) {
        y = -4 * x * x * x + 3 * x - 7;
        printf("f(%.2f) = %.2f\n", x, y);
    }
    else printf("no value\n");
    return 0;
}
```

Результати тестування програми

```
PS D:\programming\code\lab1> gcc -o main2.exe main2.c
PS D:\programming\code\lab1> .\main2.exe
Enter x: -16
no value
PS D:\programming\code\lab1> .\main2.exe
Enter x: -4
f(-4.00) = 237.00
PS D:\programming\code\lab1> .\main2.exe
Enter x: -1
no value
PS D:\programming\code\lab1> .\main2.exe
Enter x: 6
f(6.00) = 230.00
PS D:\programming\code\lab1> .\main2.exe
Enter x: 13
no value
PS D:\programming\code\lab1> .\main2.exe
Enter x: 17
f(17.00) = -19608.00
PS D:\programming\code\lab1> 
```

Висновок. На лабораторній роботі №1.1 «Розгалужені алгоритми» я засвоїв теорію та отримав практичні навички роботи з керуючими конструкціями розгалуження та логічними операціями в мові програмування C. Я виконав завдання двома способами, де другий варіант програми виявився компактнішим і вимагав менше ресурсів для виконання, ніж перший. Це дозволяє зробити висновок, що використання логічних операцій сприяє оптимізації коду, роблячи його більш ефективним.