

**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 № 1.1

з дисципліни
«Структури даних і алгоритми»

Виконав: Давидчук А.М.
студент групи ІО-41
Давидчук Артем Миколайович
номер у списку групи: 08

Перевірив:
Сергієнко А. М.

Завдання:

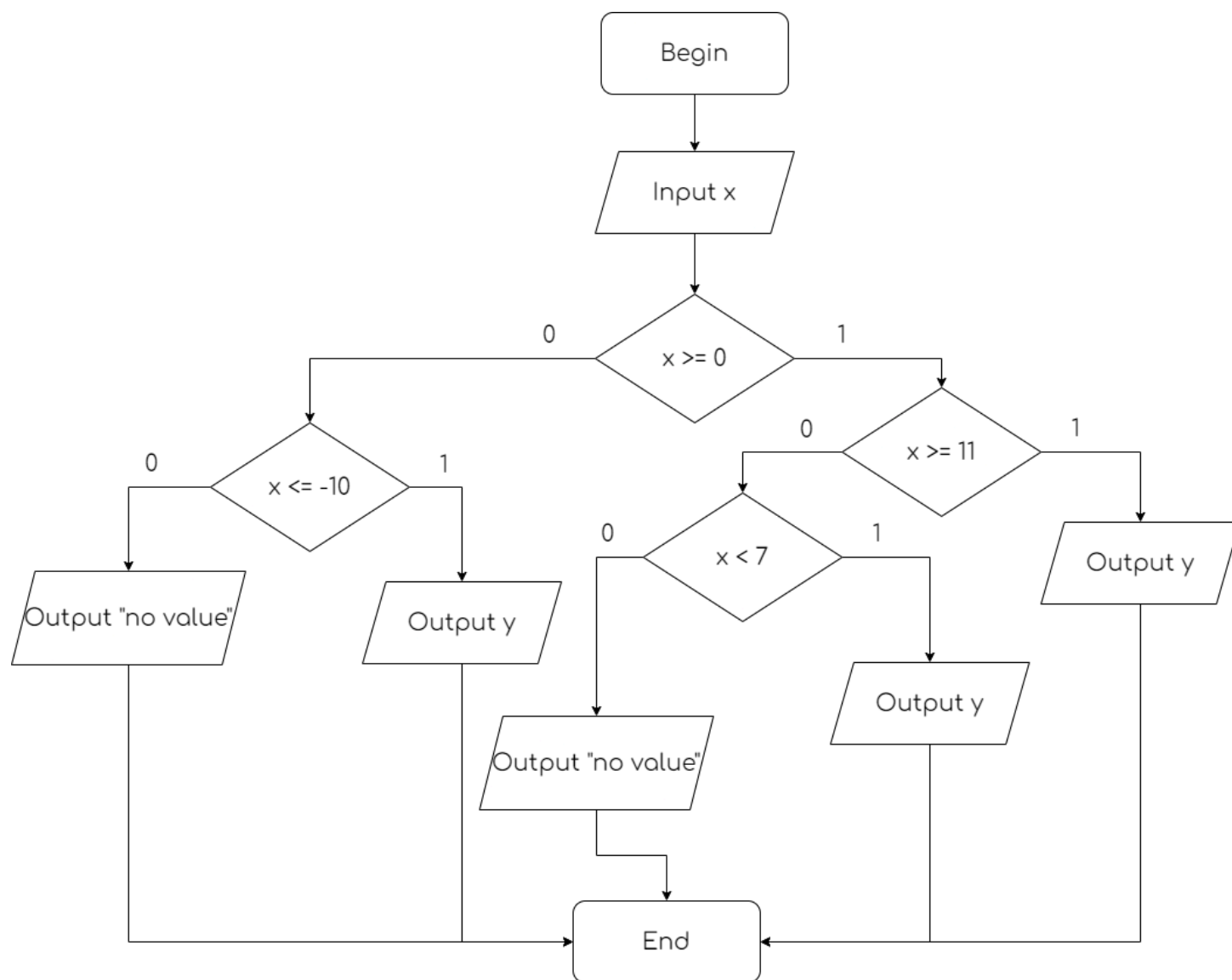
Задано дійсне число x . Визначити значення заданої за варіантом кусочно-безперервної функції $y(x)$, якщо воно існує, або вивести на екран повідомлення про неіснування функції для заданого x . Розв'язати задачу двома способами (написати дві програми):

- 1) в програмі дозволяється використовувати тільки одиничні операції порівняння ($=$, $<>$, $<$, $<=$, $>$, $>=$) і не дозволяється використовувати булеві (логічні) операції (not, and, or тощо);
- 2) в програмі необхідно обов'язково використати булеві (логічні) операції (not, and, or тощо); використання булевих операцій не повинно бути надлишковим.

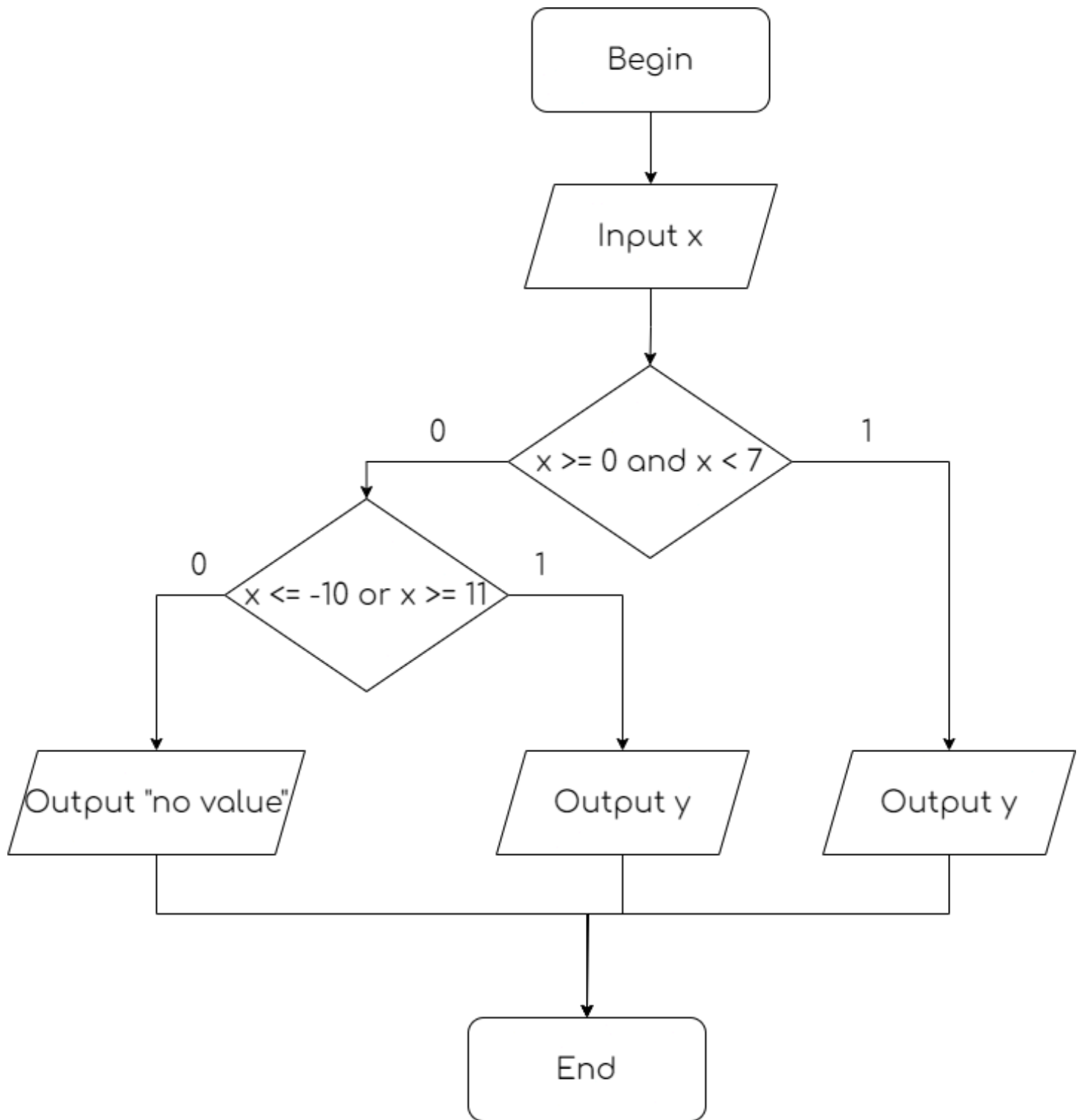
Варіант 08:

$$y = \begin{cases} -6x^2 + 8, & \text{якщо } x \in [0; 7) \\ -x^3 \div 7 + 10, & \text{якщо } x \in (-\infty; -10] \cup [11; +\infty) \end{cases}$$

Діаграма алгоритму без логічних операцій (1 алгоритм):



Діаграма алгоритму з логічними операціями (2 алгоритм):



Код алгоритму №1:

```
#include <stdio.h>
#include <math.h>

int main() {

    double x;

    printf("Enter value for x: ");
    scanf("%lf", &x);

    if (x >= 0) {

        if (x >= 11) {

            double result = -pow(x, 3) / 7 + 10;
            printf("y = %lf\n", result);
        }

        else if (x < 7) {

            double result = -6 * pow(x, 2) + 8;
            printf("y = %lf\n", result);
        }

        else {

            printf("no value\n");
        }

    }

    else if (x <= -10) {

        double result = -pow(x, 3) / 7 + 10;
        printf("y = %lf\n", result);
    }

    else {

        printf("no value\n");
    }

    return 0;

}
```

Код алгоритму №2:

```
#include <stdio.h>
#include <math.h>

int main() {

    double x;

    printf("Enter value for x: ");
    scanf("%lf", &x);

    if (x >= 0 && x < 7) {

        double result = -6 * pow(x, 2) + 8;
        printf("y = %lf\n", result);
    }

    else if (x <= -10 || x >= 11) {

        double result = -pow(x, 3) / 7 + 10;
        printf("y = %lf\n", result);
    }

    else {

        printf("no value\n");
    }

    return 0;

}
```

Результати тестування алгоритму №1:

```
Enter a value for x: 0
y = 8.000000
Enter a value for x: 7
no value
Enter a value for x: 6.999999
y = -285.999916
Enter a value for x: -12
y = 256.857143
Enter a value for x: 13
y = -303.857143
Enter a value for x: -2
no value
Enter a value for x: 4500000000000000000
y = -13017857142857141840713214936362080371609181588832649216.000000
Enter a value for x: -4500000000000000000
y = 13017857142857141840713214936362080371609181588832649216.000000
Enter a value for x: -0.0000000000000000001
no value
```

Результати тестування алгоритму №2:

```
Enter a value for x: 0
y = 8.000000
Enter a value for x: 7
no value
Enter a value for x: 6.999999
y = -285.999916
Enter a value for x: -12
y = 256.857143
Enter a value for x: 13
y = -303.857143
Enter a value for x: -2
no value
Enter a value for x: 4500000000000000000
y = -13017857142857141840713214936362080371609181588832649216.000000
Enter a value for x: -4500000000000000000
y = 13017857142857141840713214936362080371609181588832649216.000000
Enter a value for x: -0.0000000000000000001
no value
```

Висновок:

Я в першу чергу навчився програмувати алгоритми на мові C, будувати діаграми алгоритму та набув досвіду розв'язування задач алгоритмічного типу. Побудовані мною алгоритми працюють вірно та правильно з числами з плаваючою точкою, які не перевищують місткість 64 біти з урахуванням знаку та десяткової частини (для підвищення цієї межі, потрібно під'єднувати сторонні бібліотеки, що я вирішив не робити, щоб не обтяжувати код).