

**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-42  
Туров Андрій Володимирович  
номер у списку групи: 29

Перевірила:

Молчанова А. А.

Київ 2024

## Завдання

Задано дійсне число  $x$ . Визначити значення заданої за варіантом кусково-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 ( $<$ ,  $<=$ ,  $>$ ,  $>=$ ), і не дозволяється використовувати булеві (логічні) операції ( $!$ ,  $\&\&$ ,  $\parallel$  тощо);

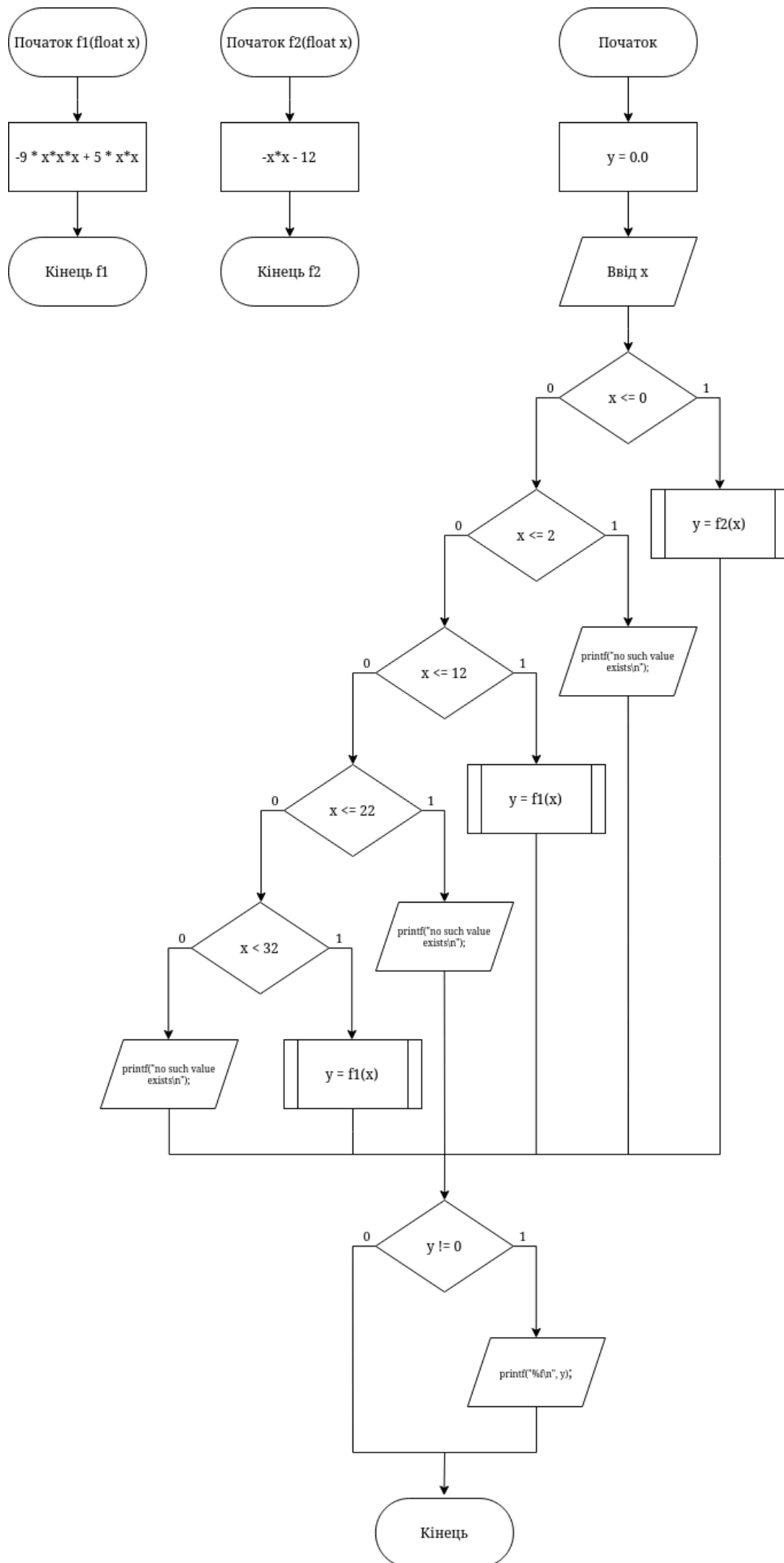
2) у програмі необхідно обов'язково використати булеві (логічні) операції ( $!$ ,  $\&\&$ ,  $\parallel$  тощо); використання булевих операцій не повинно бути надлишковим.

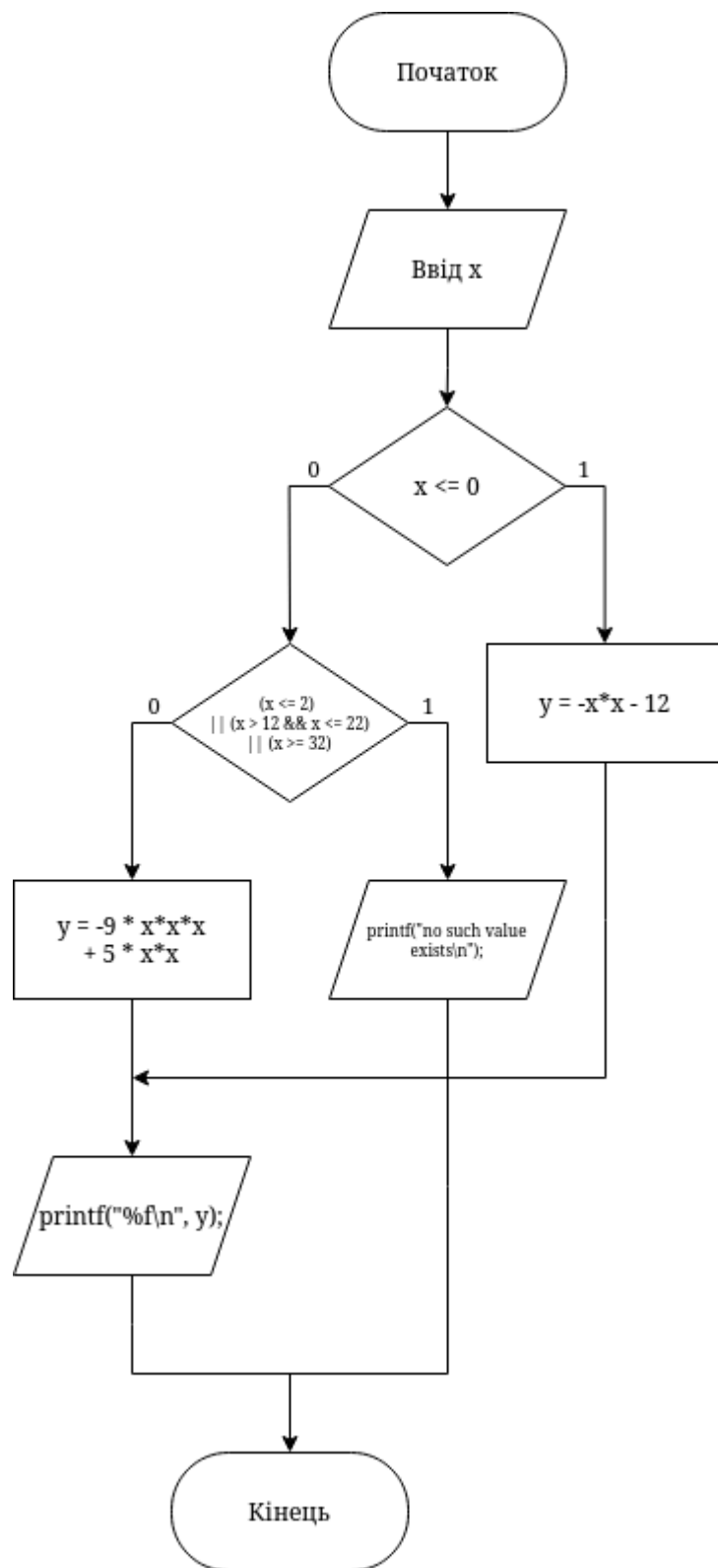
## Варіант 29

$$y = -9x^3 + 5x^2, \quad x \in (2; 12] \cup (22; 32)$$

$$y = -x^2 - 12, \quad x \in (-\infty; 0]$$

## Діаграма алгоритму





### Текст програми

A)

```

// одиничні операції порівняння без логічних операцій
// y = -9x^3 + 5x^2          (2; 12] U (22, 32)
//   = -x^2 - 12           (-inf, 0]

```

```

#include <stdio.h>

float f1(float x) {
    return -9 * x*x*x + 5 * x*x;
}

float f2(float x) {
    return -x*x - 12;
}

int main(void) {
    float x;
    float y = 0; // y can never be 0.

    scanf("%f", &x);

    if (x <= 0) {
        y = f2(x);
    } else if (x <= 2) {
        printf("no such value exists\n");
    } else if (x <= 12) {
        y = f1(x);
    } else if (x <= 22) {
        printf("no such value exists\n");
    } else if (x < 32) {
        y = f1(x);
    } else {
        printf("no such value exists\n");
    }

    if (y != 0) {
        printf("%f\n", y);
    }
}

```

```
        return 0;
    }
}
```

B)

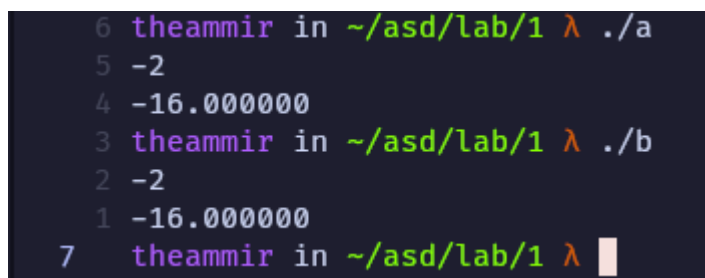
```
// обов'язкові логічні операції, не мають бути надлишковими
//  $y = -9x^3 + 5x^2$  (2; 12] U (22, 32)
//  $y = -x^2 - 12$  (-inf, 0]
#include <stdio.h>
int main() {
    float x, y;

    scanf("%f", &x);

    if (x <= 0) {
        y = -x*x - 12;
    } else if (x <= 2 || (x > 12 && x <= 22) || (x >= 32)) {
        printf("no such value exists\n");
        return 0;
    } else {
        y = -9 * x*x*x + 5 * x*x;
    }

    printf("%f\n", y);
    return 0;
}
```

### Результати тестування програми



```
6 theamir in ~/asd/lab/1 λ ./a
5 -2
4 -16.000000
3 theamir in ~/asd/lab/1 λ ./b
2 -2
1 -16.000000
7 theamir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 2
4 no such value exists
3 theammir in ~/asd/lab/1 λ ./b
2 2
1 no such value exists
7 theammir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 5
4 -1000.000000
3 theammir in ~/asd/lab/1 λ ./b
2 5
1 -1000.000000
7 theammir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 12
4 -14832.000000
3 theammir in ~/asd/lab/1 λ ./b
2 12
1 -14832.000000
7 theammir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 15.888
4 no such value exists
3 theammir in ~/asd/lab/1 λ ./b
2 15.7
1 no such value exists
7 theammir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 22
4 no such value exists
3 theammir in ~/asd/lab/1 λ ./b
2 22
1 no such value exists
7 theammir in ~/asd/lab/1 λ
```

```
6 theammir in ~/asd/lab/1 λ ./a
5 25.1
4 -139169.218750
3 theammir in ~/asd/lab/1 λ ./b
2 25.1
1 -139169.218750
7 theammir in ~/asd/lab/1 λ
```

## **Висновки**

Під час виконання лабораторної роботи використав мову програмування C для задання кускової функції й обчислення її значення у точці за допомогою конструкцій розгалуження (`if-else`).

Використав на практиці складніші логічні вирази, отримавши в результаті компактніший код, таким чином навчився користуватись булевими операціями, звернув увагу також на пріоритет виконання цих операторів.

Додатково виніс куски алгебраїчної функції у власні підпрограми, оскільки в коді **A** вони використовуються декілька разів. Програмісту в подальшому буде (трохи) легше змінити логіку виконання, бо вона визначається на початку програми лише один раз.