

## 1. Постановка задачі

1. Для кожної задачі зобразити алгоритм розв'язку у вигляді блок-схеми за допомогою програмного засобу уEd Graph Editor. Блок-схему додати до звіту.
2. Для кожної задачі написати на мові C++ програму (консольний застосунок).  
Налагодити і протестувати написану програму.  
Додати текст програми до звіту.  
Додати до звіту описання тестових прикладів і інтерфейсу програми (тобто що мусить робити користувач, щоби програма виконала свою функцію).
3. При виконанні завдань не використовувати глобальні змінні, математичні функції, масиви, string.  
Кількість тестових прикладів повинна демонструвати правильність роботи програми.

### **SUM**

Завдання SUM

=====

Скласти програму SUM, яка:

- 1) вводить з клавіатури послідовність цілих чисел, що закінчується нулем;
- 2) знаходить суму всіх членів послідовності;
- 3) виводить на екран знайдене значення у вигляді:  
"SUM=...".

Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

a[1]= 13

a[2]= 21

a[3]= 2

a[4]= 14

a[5]= -1

a[6]= -2

a[7]= 0

SUM=47

## PRODUCT

Завдання PRODUCT

=====

Скласти програму PRODUCT, яка:

- 1) вводить з клавіатури послідовність цілих чисел, що закінчується нулем;
- 2) знаходить добуток усіх парних членів послідовності;
- 3) виводить на екран обчислений добуток у вигляді: "ДОБУТОК = ...".

Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

a[1]= 13

a[2]= 21

a[3]= 2

a[4]= 14

a[5]= -1

a[6]= -2

a[7]= 0

ДОБУТОК=-56

## COUNTODD

Задача "CountOdd"

=====

Скласти програму CountOdd, яка:

1. Вводить із клавіатури послідовність цілих чисел, що закінчується нулем.
2. Знаходить скільки серед них непарних.
3. Як тільки вводиться число 0, програма припиняє роботу та видає результат підрахунку на екран.

Приклад :

Введіть число = 11

Введіть число = 10

Введіть число = 3

Введіть число = 1

Введіть число = 2

Введіть число = 0

Зустрілося 3 непарних числа.

## Parrot2

Завдання Parrot2

=====

Складіть програму PARROT2, яка:

1. Просить користувача ввести з клавіатури один рядок якогось тексту.

2. Виводить цей рядок на екран.

3. Повторює ці дії до того часу, поки користувач не введе порожній рядок (тобто не натисне просто ENTER).

## **MENU2**

Завдання MENU2

=====

Скласти програму MENU2, яка:

1) виводить на екран наступний текст:

"Працює інформаційна система ФПМ:

1. Новини дня

2. Анекдот тижня

3. Автор!

0. Вихід

Введіть відповідний номер та натисніть ENTER."

2) вводять із клавіатури номер;

3) виводить на екран відповідну інформацію;

4) продовжує циклічно виконувати вищезгадані дії доки користувач не вибере "Вихід".

## **MAX**

Завдання MAX

=====

Скласти програму MAX, яка:

1) вводять з клавіатури послідовність цілих чисел, яка завершується нулем;

2) знаходить найбільший із усіх членів послідовності;

3) виводить на екран знайдене значення у вигляді: "MAX = ...".

Приклад:

Введіть послідовність цілих чисел. Нуль – ознака кінця.

a[1]= 13

a[2]= 21

a[3]= 2

a[4]= 14

a[5]= -1

a[6]= -2

a[7]= 0

MAX=21

## **Fibonacci.txt**

Завдання Fibonacci (FIBO)

=====

Послідовність чисел Фібоначчі - це числа f0, f1, f2, f3 ... :

f[0]=0

f[1]=1

f[N]=f[N-1]+F[N-2], якщо N>=2.

Легко бачити, що з цього визначення виходить наступна послідовність чисел:

0 1 1 2 3 5 8 13 21 ...

Складіть програму FIBO, яка вводять з клавіатури число N, обчислює число Фібоначчі з номером N (тобто f[N]) та виводить його на екран.

Приклад роботи програми:

N = 7

$$F(7) = 13$$

## **НІТ2**

### **Завдання НІТ2**

=====

Скласти програму НІТ2, яка:

0. "Задумує" коло  $O$  з центром у точці  $(x_0, y_0)$  та радіусом  $R$ .

1. Вводить з клавіатури координати  $(x, y)$  точки  $A$  на площині.

2. Перевіряє, чи потрапляє точка  $A$  всередину (або на кордон) кола  $O$ .

3. Доки точка не потрапила всередину, виводить повідомлення "Тепліше" або "Холодніше" в залежності від того, ближче або далі від центру кола потрапила нова точка порівняно з попередньою, та повторює пункти 1, 2, 3.

4. Виводить результат на екран у вигляді:

"Точка  $(x, y)$  потрапила в коло з центром у точці  $(x_0, y_0)$  радіуса  $R$ ", підставляючи результат замість  $R, x_0, y_0, x, y$  їх значення.

## **Factor.txt**

### **Завдання Factor**

=====

Скласти програму FACTOR, яка для заданого натурального числа  $N$  обчислює факторіал цієї кількості.

Факторіал  $N$  позначається так:  $N!$

За визначенням  $N! = 1 * 2 * 3 * 4 * \dots * (N-1) * N$

Наприклад:  $4! = 1 * 2 * 3 * 4 = 24$

Приклад:

Для обчислення  $N!$  введіть:

$N = 4$

$4! = 24$

2) Скласти програму FACTOR2 – удосконалений варіант програми FACTOR.

Ця програма у відповідь на неправильно введені  $N$  (тобто не натуральне число) видає зрозумілу підказку користувачеві.

Приклад:

$N = -1$

Error:  $N$  мусить бути додатнім, не negative!

Приклад:

$N = 3.14$

Error:  $N$  must be natural, not real!

Приклад:

$N = \text{шість}$

Error:  $N$  must a number, no a string!

## **2. Опис розв'язку + блок-схема**

Опис розв'язку:

Програма складається з дев'ятох основних функцій, кожна з яких вирішує одне з шести завдань або виводить меню для користувача.

Кожна функція починається з виведення відомостей про завдання та зчитує необхідні дані від користувача (наприклад, числа, координати тощо).

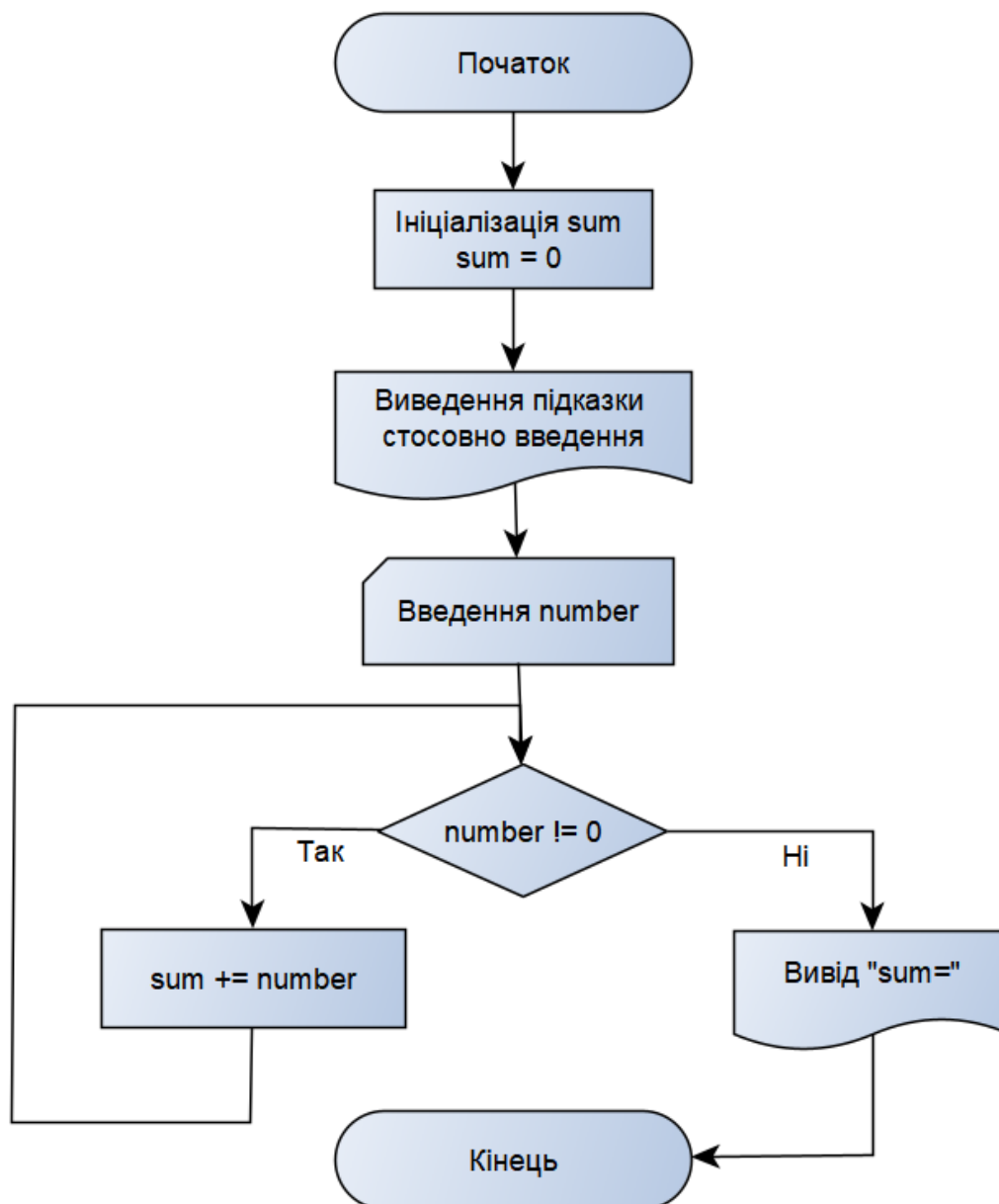
Після цього виконуються відповідні обчислення чи операції відповідно до завдання.

Результат виводиться на екран.

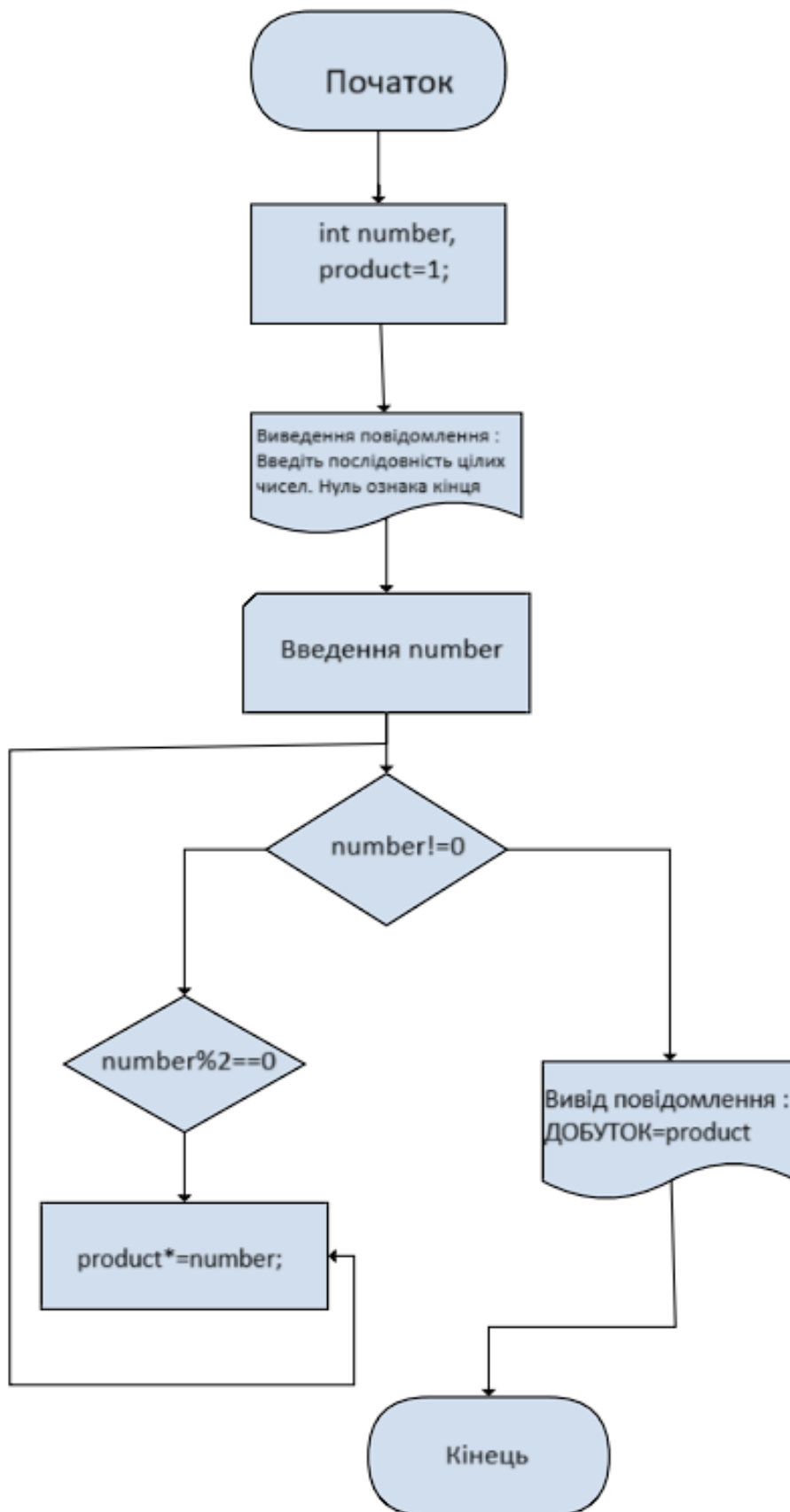
Програма повертається до головного меню для можливості вибору іншого завдання.

Блок-схема:

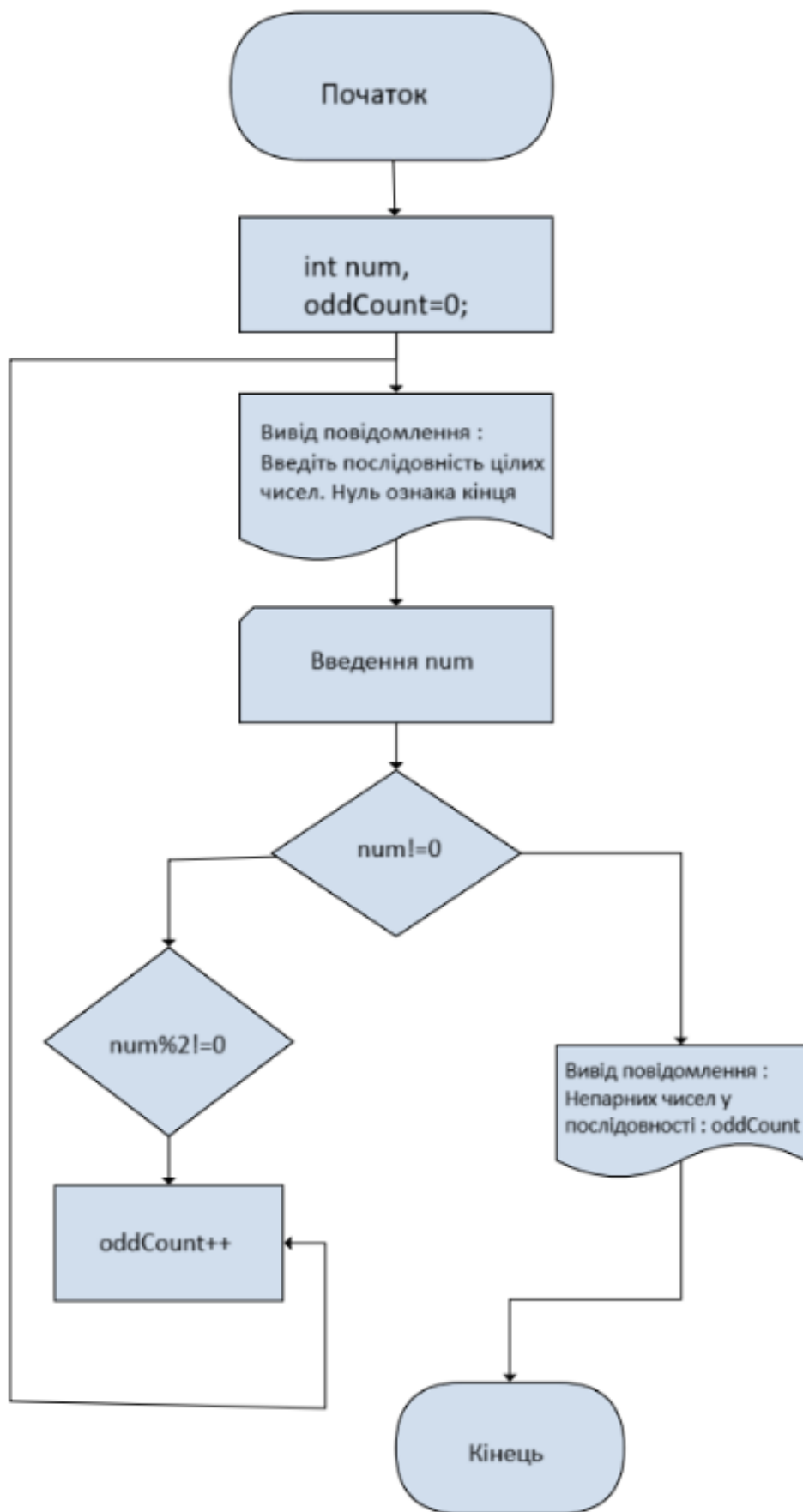
1 завдання:



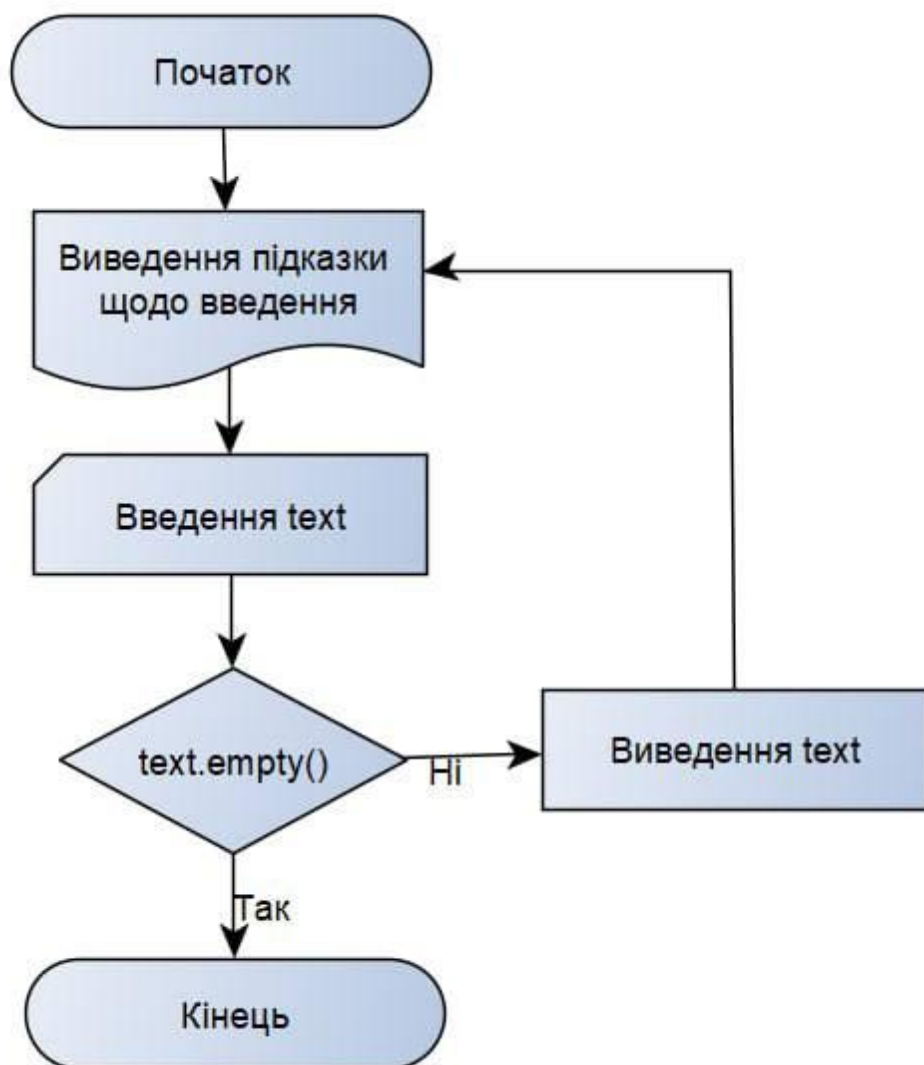
2 завдання:



3 завдання:

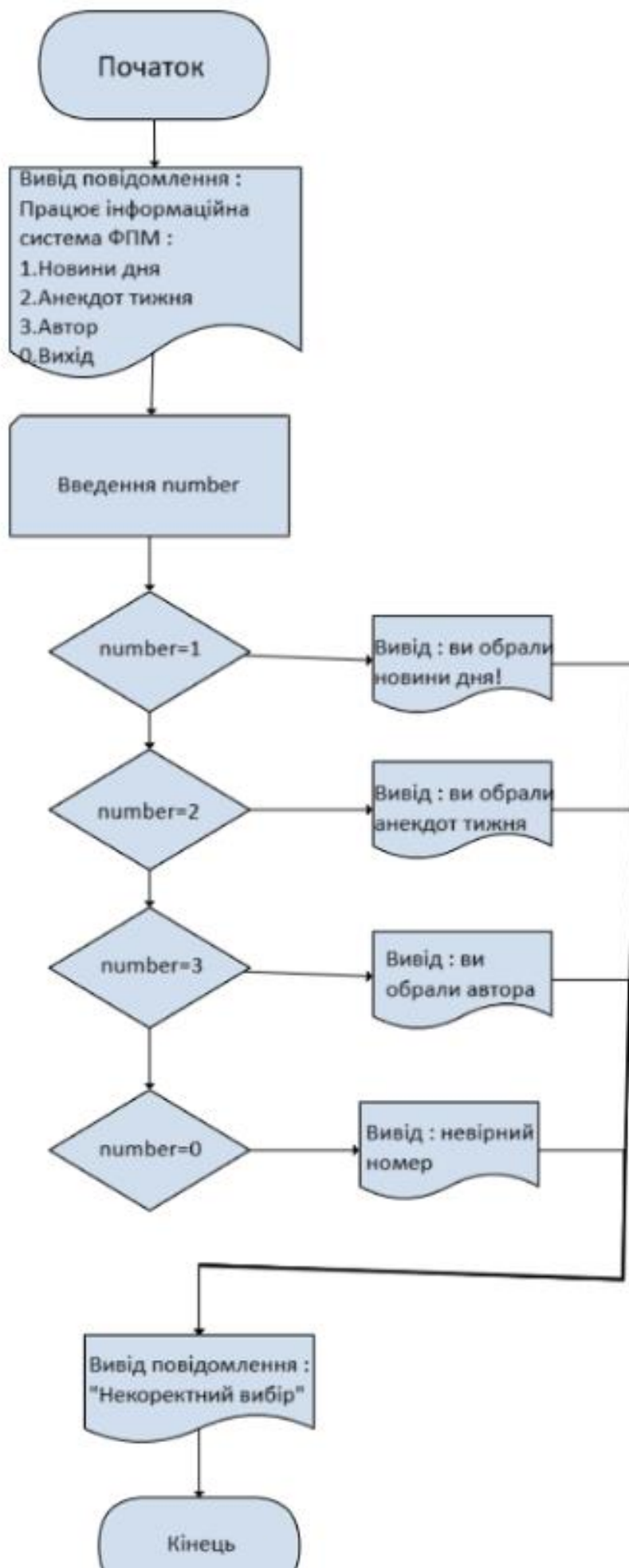


4 завдання:

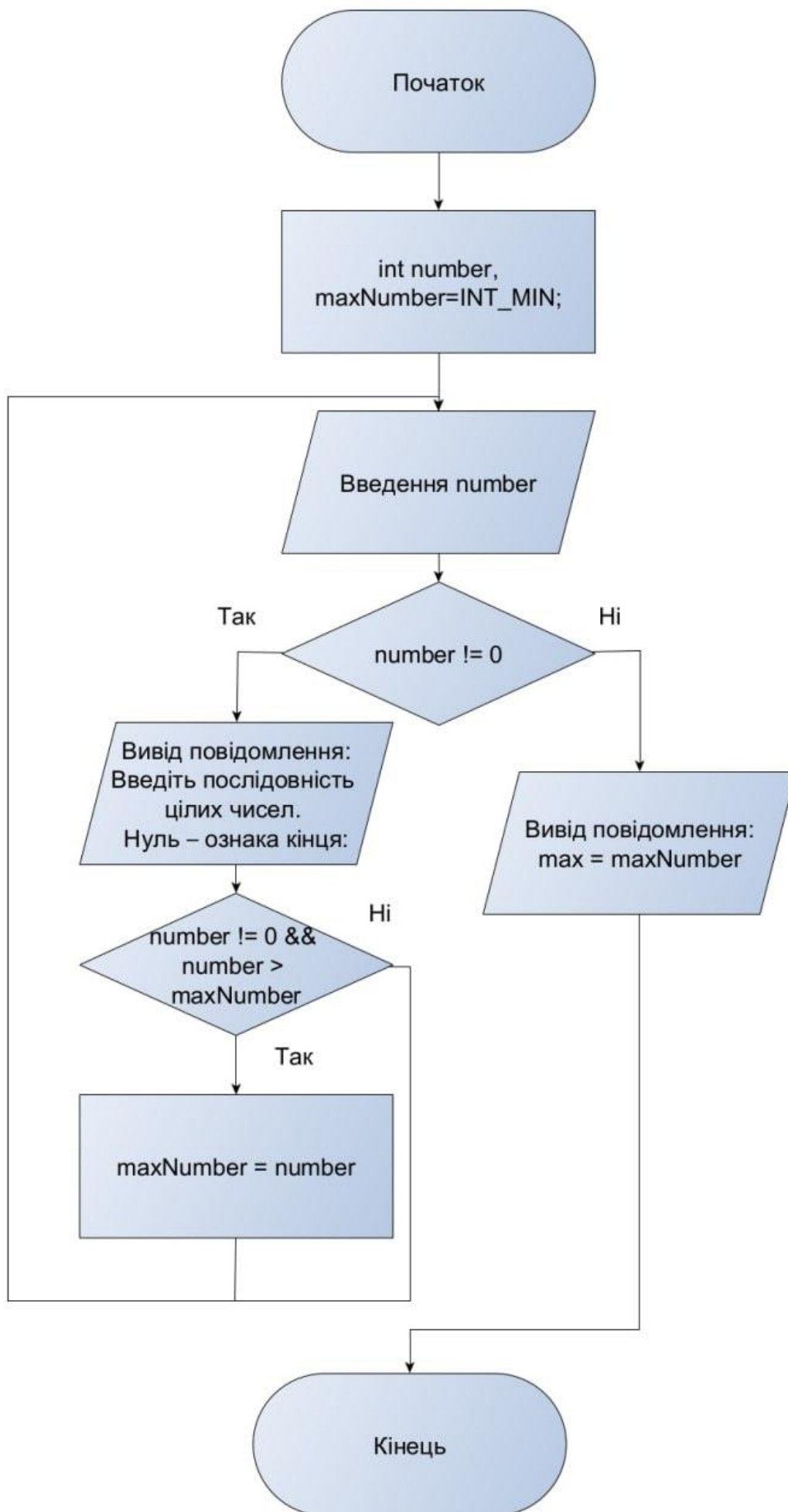




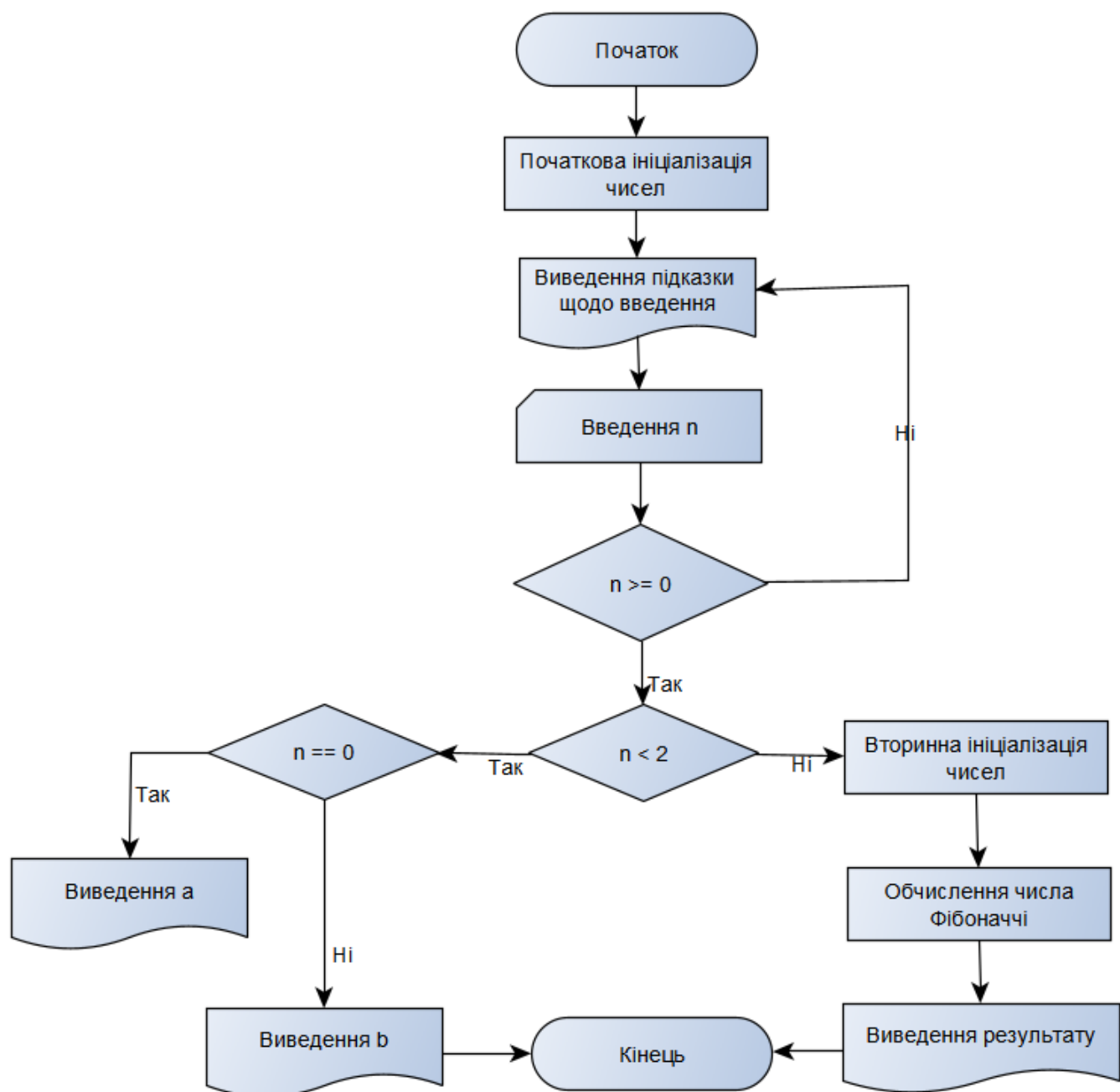
5 завдання:



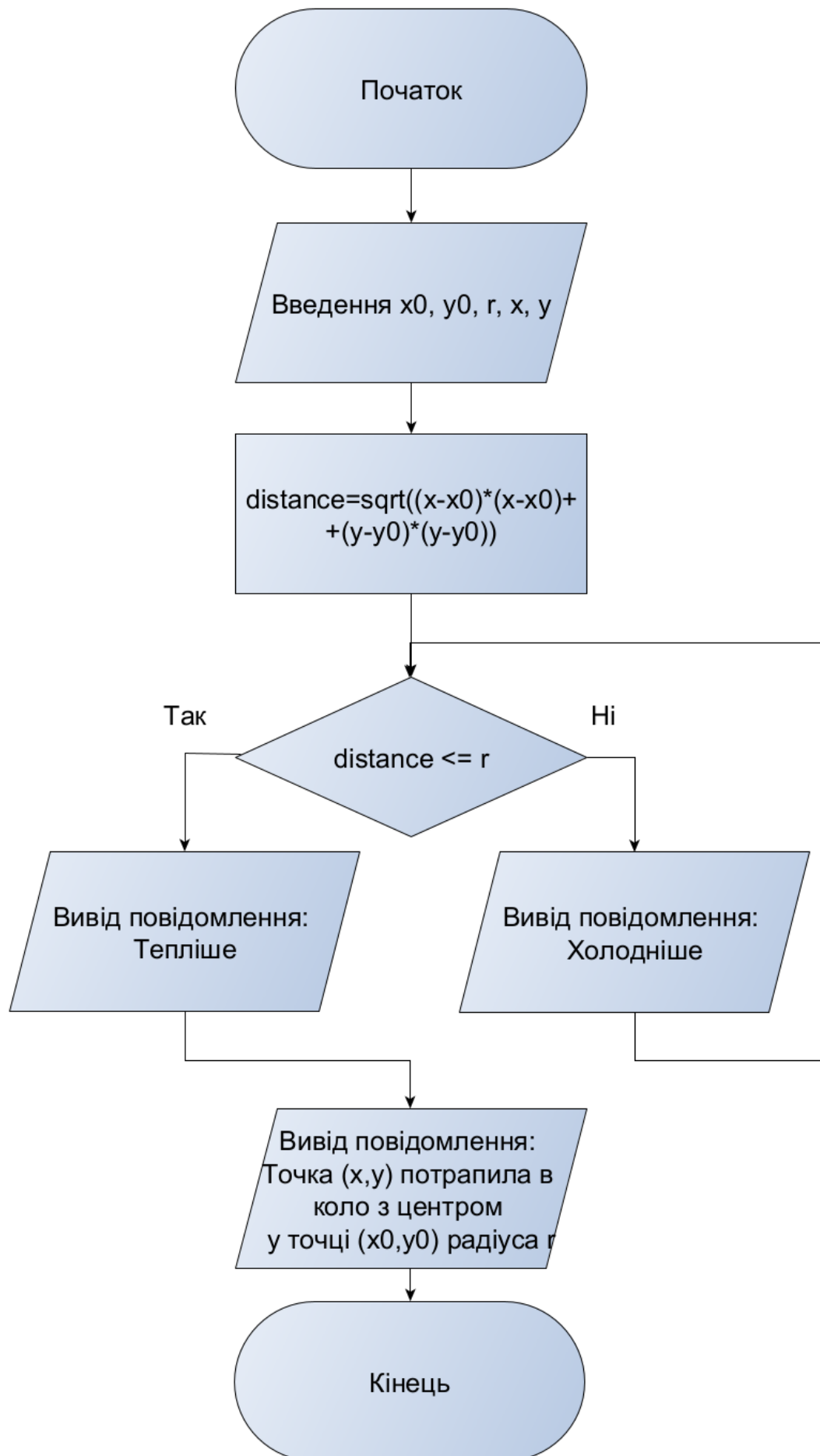
6 завдання:



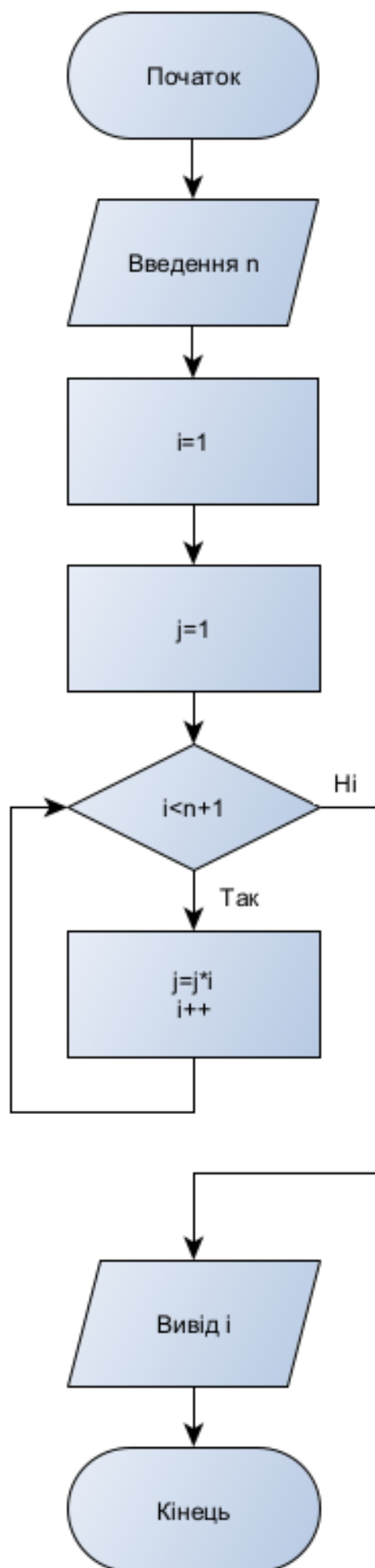
7 завдання:



8 завдання:



9 завдання:



### 3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

Усі завдання в одному коді

**Код:**

```
#include <iostream>

#include <cmath>

#include <limits>

#include <climits>

using namespace std;

int main() {

    int choice;

    // Виведення меню

    cout << "Виберіть завдання:" << endl;

    cout << "1. SUM" << endl;

    cout << "2. PRODUCT" << endl;

    cout << "3. COUNTODD" << endl;

    cout << "4. Parrot2" << endl;

    cout << "5. MENU2" << endl;

    cout << "6. MAX" << endl;

    cout << "7. Fibonacci.txt" << endl;

    cout << "8. HIT2" << endl;

    cout << "9. Factor.txt" << endl;

    cout << "Введіть відповідний номер і натисніть ENTER: ";

    cin >> choice;

    switch (choice) {

        case 1: {

            int number, sum = 0;

            cout << "Введіть послідовність цілих чисел. Нуль – ознака кінця:";

            do {

                cin >> number;

                sum += number;

            } while (number != 0);
```

```
    cout << "sum = " << sum << endl;
    break;
}
```

```
case 2: {
```

```
int number= 1;
int r = 0;
int product =1;
cout << "Введіть послідовність цілих чисел. Нуль – ознака кінця: ";
while (true) {
    cin >> number;
    if (number == 0) {
        break;
    }
    if (number % 2 == 0) {
        product *= number;r++;
    }
}
if (r!=0) cout << "ДОБУТОК = " << product << endl;
else cout << "Нема парних чисел" << endl;
break;
}
```

```
case 3: {
```

```
int num, oddCount = 0;
cout << "Введіть послідовність цілих чисел. Нуль – ознака кінця: " << endl;
while (true) {
    cin >> num;

    if (num == 0) {
```

```
        break;
    }
```

```
    if (num % 2 != 0) {
        oddCount++;
    }
}
cout << "Непарних чисел у послідовності: " << oddCount << endl;
break;
}
```

```
case 4: {
```

```
    char text[1000];
    cout << "Введіть рядок тексту: ";
    cin.ignore(32767, '\n');
    do {
        cin.getline(text, 1000);

        if (text[0] != '\0') {
            cout << "Введений рядок: " << text << endl;
            cout << "Введіть рядок тексту: ";
        }
    } while (text[0] != '\0');
```

```
    cout << "Програма завершена." << endl;
    break;
}
```

```
case 5: {
```

```
    int number;
```



```

while (true) {
    cout << "Працює інформаційна система ФПМ:\n";
    cout << "1. Новини дня\n";
    cout << "2. Анекдот тижня\n";
    cout << "3. Автора!\n";
    cout << "0. Вихід\n";
    cout << "Введіть відповідний номер та натисніть enter: ";
    cin >> number;
    switch (number) {
        case 1:
            cout << "Ви обрали Новини дня.\n";
            break;
        case 2:
            cout << "Ви обрали Анекдот тижня.\n";
            break;
        case 3:
            cout << "Ви обрали Автора!\n";
            break;
        case 0:
            cout << "Програма завершена!\n";
            return 0;
        default:
            cout << "Невірний номер. Будь ласка, спробуйте ще раз.\n";
            break;
    }
    cout << endl;
}
break;
}

```

```

case 6: {

```

```

int number;

int maxNumber = INT_MIN; // початкове значення максимального числа
cout << "Введіть послідовність цілих чисел. Нуль – ознака кінця:" << endl;
do {
    cin >> number;

    if (number != 0 && number > maxNumber) {
        maxNumber = number;
    }
} while (number != 0);

cout << "max = " << maxNumber << endl;
break;
}

```

```

case 7: {
    int n, t;
    int a = 0;
    int b = 1;
    do {
        cout << "Введіть номер числа Фібоначчі: ";
        cin >> n;
        if (n >= 0) {
            if (n < 2) {
                if (n == 0) cout << "Число Фібоначчі з номером " << 0 << ": " << a << endl;
                else cout << "Число Фібоначчі з номером " << 1 << ": " << b << endl;
            } else {
                int originalA = 0;
                int originalB = 1;
                for (int i = 2; i <= n; i++) {
                    int temp = a + b;
                    a = b;
                    b = temp;
                }
                cout << "Число Фібоначчі з номером " << n << ": " << b << endl;
                a = originalA;
                b = originalB;
            }
        } else {
            cout << "Число має бути додатне!" << endl;
        }
    } while (n < 0);
}

```

```

        break;
    }
    case 8: {

        double x0, y0; // Координати центра кола
        double r; // Радіус кола
        cout << "Введіть координати центра кола (x0, y0): ";
        cin >> x0 >> y0;
        cout << "Введіть радіус кола r: ";
        cin >> r;
        double x, y; // Координати точки a
        do {
            cout << "Введіть координати точки a (x, y): ";
            cin >> x >> y;

            double distance = sqrt((x - x0) * (x - x0) + (y - y0) * (y - y0)); // Відстань від точки
            до центра кола

            if (distance <= r) {
                cout << "Точка (" << x << ", " << y << ") потрапила в коло з центром у точці ("
                << x0 << ", " << y0 << ") радіуса " << r << endl;
                break;
            }
            else if (distance < r) {
                cout << "Тепліше" << endl;
            }
            else {
                cout << "Холодніше" << endl;
            }
        } while (true);
        break;
    }

```

```
case 9: {
```

```
    int subChoice;
```

```
    cout << "Виберіть тип факторіалу:" << endl;
```

```
    cout << "1. FACTOR" << endl;
```

```
    cout << "2. FACTOR2" << endl;
```

```
    cout << "Введіть відповідний номер і натисніть ENTER: ";
```

```
    cin >> subChoice;
```

```
    switch (subChoice) {
```

```
        case 1: {
```

```
            int n;
```

```
            cout << "Введіть число: ";
```

```
            cin >> n;
```

```
            int j = 1;
```

```
            for (int i = 1; i < n + 1; i++) j = j * i;
```

```
            cout << "Факторіал числа " << n << ": " << j << endl;
```

```
            break;
```

```
        }
```

```
case 2: {
```

```
    int N;
```

```
    while (true) {
```

```
        std::cout << "Введіть число N: ";
```

```
        if (!(std::cin >> N) || cin.peek() != '\n') {
```

```
            std::cout << "Помилка: N мусить бути натуральним числом, не рядком і не  
дійсним числом!" << std::endl;
```

```
            std::cin.clear();
```

```
            std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
```

```

    } else if (N <= 0) {
        std::cout << "Помилка: N мусить бути додатнім числом, не від'ємним!" <<
std::endl;
    } else {
        // Виконати операції з N
        std::cout << "Введено правильне число N: " << N << std::endl;
        int j = 1;
        for (int i = 1; i < N + 1; i++) {
            j = j * i;
        }
        std::cout << "Факторіал числа " << N << ": " << j << std::endl;
        break;
    }
}

}

return 0;
}

}

}

```

#### 4. Опис інтерфейсу програми (керівництво користувача)

Загальна програма складається з меню, яке дозволяє користувачеві вибрати одну з дев'яти опцій. Кожна опція відповідає одному з дев'яти завдань. Нижче наведено опис інтерфейсу та керівництва користувача:

Інструкція користувача

Вибір завдання

Відкрийте програму.

У вас з'явиться меню з шістьма опціями:

Виберіть завдання:

1. SUM

2. PRODUCT

3. COUNTODD

4. Parrot2

5. MENU2

6. MAX

7. Fibonacci.txt

8. HIT2

9. Factor.txt

Введіть відповідний номер і натисніть ENTER:

Оберіть завдання

Введіть номер завдання, яке ви хочете виконати (від 1 до 9).

Натисніть клавішу ENTER.

Виконання завдання

Залежно від вибору, програма виконає одне з наступних:

SUM (Завдання 1)

Введіть з клавіатури послідовність цілих чисел, що закінчується нулем.

Програма обчислить суму всіх членів послідовності та виведе результат.

PRODUCT (Завдання 2)

Введіть з клавіатури послідовність цілих чисел, що закінчується нулем.

Програма обчислить добуток усіх парних членів послідовності та виведе результат.

COUNTODD (Завдання 3)

Введіть з клавіатури послідовність цілих чисел, що закінчується нулем.

Програма знаходить скільки серед них непарних членів послідовності та виведе результат.

Parrot2 (Завдання 4)

Введіть з клавіатури один рядок якогось тексту.

Програма виводить цей рядок на екран і повторює ці дії до того часу, поки користувач не введе порожній рядок (тобто не натисне просто ENTER).

MENU2 (Завдання 5)

Введіть номер та натисніть ENTER.

Програма виводить на екран відповідну інформацію і продовжує циклічно виконувати вищезгадані дії доки користувач не вибере "Вихід".

#### MAX (Завдання 6)

Введіть з клавіатури послідовність цілих чисел, що закінчується нулем.

Програма знаходить найбільший із усіх членів послідовності та виведе результат.

#### Fibonacci.txt (Завдання 7)

Введіть номер N числа Фібоначчі.

Програма обчислює число Фібоначчі з номером N та виводить його на екран.

#### НІТ2 (Завдання 8)

Введіть координати центра кола  $x_0$  та  $y_0$ , радіус  $R$ , та координати точки  $x$  та  $y$ .

Програма перевіряє, чи потрапляє точка А всередину (або на кордон) кола О. Доки точка не потрапила всередину, виводить повідомлення "Тепліше" або "Холодніше" в залежності від того, ближче або далі від центру кола потрапила нова точка порівняно з попередньою.

#### Factor.txt (Завдання 9)

Введіть номер N.

Програма обчислює для заданого натурального числа N обчислює факторіал цієї кількості.

## 5. Опис тестових прикладів

### Завдання SUM

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 1
Введіть послідовність цілих чисел. Нуль - ознака кінця:5
6
0
sum = 11

Process returned 0 (0x0)   execution time : 10.395 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 0

Process returned 0 (0x0)   execution time : 5.725 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 1
Введіть послідовність цілих чисел. Нуль - ознака кінця:6
9
7
3
0
sum = 25

Process returned 0 (0x0)   execution time : 8.261 s
Press any key to continue.
```

## Завдання PRODUCT

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 2
Введіть послідовність цілих чисел. Нуль - ознака кінця: 8
2
0
ДОБУТОК = 16

Process returned 0 (0x0)   execution time : 7.378 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 2
Введіть послідовність цілих чисел. Нуль - ознака кінця: 0
ДОБУТОК = 0

Process returned 0 (0x0)   execution time : 12.873 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 2
Введіть послідовність цілих чисел. Нуль - ознака кінця: 8
1
0
ДОБУТОК = 8

Process returned 0 (0x0)   execution time : 12.241 s
Press any key to continue.
```

## Завдання COUNTODD

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 3
Введіть послідовність цілих чисел. Нуль - ознака кінця:
4
7
0
Непарних чисел у послідовності: 1

Process returned 0 (0x0)   execution time : 9.418 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 3
Введіть послідовність цілих чисел. Нуль - ознака кінця:
8
2
0
Непарних чисел у послідовності: 0

Process returned 0 (0x0)   execution time : 6.280 s
Press any key to continue.
```



```
Введіть відповідний номер і натисніть ENTER: 3
Введіть послідовність цілих чисел. Нуль - ознака кінця:
0
Непарних чисел у послідовності: 0

Process returned 0 (0x0)   execution time : 4.638 s
Press any key to continue.
```

## Завдання Parrot2

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 4
Введіть рядок тексту: іукр
Введений рядок: іукр
Введіть рядок тексту: 0
Введений рядок: 0
Введіть рядок тексту:
Введіть рядок тексту: Програма завершена.

Process returned 0 (0x0)   execution time : 10.597 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 4
Введіть рядок тексту: сн
Введений рядок: сн
Введіть рядок тексту: р
Введений рядок: р
Введіть рядок тексту:
Введіть рядок тексту: Програма завершена.

Process returned 0 (0x0)   execution time : 6.319 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 4
Введіть рядок тексту:
Введіть рядок тексту: Програма завершена.

Process returned 0 (0x0)   execution time : 3.407 s
Press any key to continue.
```

## Завдання MENU2

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 5
Працює інформаційна система ФПМ:
1. Новини дня
2. Анекдот тижня
3. Автора!
0. Вихід
Введіть відповідний номер та натисніть enter: 1
Ви обрали Новини дня.

Працює інформаційна система ФПМ:
1. Новини дня
2. Анекдот тижня
3. Автора!
0. Вихід
Введіть відповідний номер та натисніть enter: 0
Програма завершена!

Process returned 0 (0x0)   execution time : 7.404 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 5
Працює інформаційна система ФПМ:
1. Новини дня
2. Анекдот тижня
3. Автора!
0. Вихід
Введіть відповідний номер та натисніть enter: 0
Програма завершена!

Process returned 0 (0x0)   execution time : 5.628 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 5
Працює інформаційна система ФПМ:
1. Новини дня
2. Анекдот тижня
3. Автора!
0. Вихід
Введіть відповідний номер та натисніть enter: 2
Ви обрали Анекдот тижня.

Працює інформаційна система ФПМ:
1. Новини дня
2. Анекдот тижня
3. Автора!
0. Вихід
Введіть відповідний номер та натисніть enter: 0
Програма завершена!

Process returned 0 (0x0)   execution time : 7.433 s
Press any key to continue.
```

## Завдання MAX

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 6
Введіть послідовність цілих чисел. Нуль - ознака кінця:
0
max = 0

Process returned 0 (0x0)   execution time : 5.265 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 6
Введіть послідовність цілих чисел. Нуль - ознака кінця:
8
0
max = 8

Process returned 0 (0x0)   execution time : 6.132 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 6
Введіть послідовність цілих чисел. Нуль - ознака кінця:
4
111
0
max = 111

Process returned 0 (0x0)   execution time : 8.634 s
Press any key to continue.
```

## Завдання Fibonacci.txt

## Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 7
Введіть номер числа Фібоначчі: 0
Число Фібоначчі з номером 0: 0

Process returned 0 (0x0)   execution time : 3.969 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 7
Введіть номер числа Фібоначчі: 1
Число Фібоначчі з номером 1: 1

Process returned 0 (0x0)   execution time : 3.973 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 7
Введіть номер числа Фібоначчі: 6
Число Фібоначчі з номером 6: 8
```

## Завдання НІТ2

### Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 8
Введіть координати центра кола (x0, y0): 0
0
Введіть радіус кола r: 1
Введіть координати точки a (x, y): 0
0
Точка (0, 0) потрапила в коло з центром у точці (0, 0) радіуса 1

Process returned 0 (0x0)   execution time : 13.812 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 8
Введіть координати центра кола (x0, y0): 1
1
Введіть радіус кола r: 1
Введіть координати точки a (x, y): 2
2
Холодніше
Введіть координати точки a (x, y): 2
1
Точка (2, 1) потрапила в коло з центром у точці (1, 1) радіуса 1

Process returned 0 (0x0)   execution time : 16.599 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 8
Введіть координати центра кола (x0, y0): 1
2
Введіть радіус кола r: 3
Введіть координати точки a (x, y): 2
2
Точка (2, 2) потрапила в коло з центром у точці (1, 2) радіуса 3

Process returned 0 (0x0)   execution time : 12.001 s
Press any key to continue.
```

## Завдання Factor.txt

Приклад роботи програми :

```
Введіть відповідний номер і натисніть ENTER: 9
Введіть число: 6
Факторіал числа 6: 720

Process returned 0 (0x0)   execution time : 4.911 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 9
Введіть число: 3
Факторіал числа 3: 6

Process returned 0 (0x0)   execution time : 3.772 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 9
Введіть число: 5
Факторіал числа 5: 120

Process returned 0 (0x0)   execution time : 7.186 s
Press any key to continue.
```

```
Введіть відповідний номер і натисніть ENTER: 9
Виберіть тип факторіалу:
1. FACTOR
2. FACTOR2
Введіть відповідний номер і натисніть ENTER: 2
Введіть число N: 5
Введено правильн❖ число N: 5
```

```
Введіть відповідний номер і натисніть ENTER: 9
Виберіть тип факторіалу:
1. FACTOR
2. FACTOR2
Введіть відповідний номер і натисніть ENTER: 2
Введіть число N: три
Помилка: N мусить ❖ти числом, не рядком!
```

## 6. Аналіз помилок (опис усунення зауважень)

Ось декілька помилок з котрими я стискалась:

### Помилка вводу даних:

Помилка: Користувач вводить некоректні дані (наприклад, нечислові значення, літери замість чисел).

Виправлення: Додати перевірку введених даних та повідомлення про помилку. Можна використовувати цикл для повторного введення даних, поки користувач не введе коректні значення.

### Невірна обробка винятків:

Помилка: Не передбачено випадки непередбачених ситуацій, які можуть призвести до аварійного завершення програми.

Виправлення: Додати блоки обробки винятків для виявлення та вирішення непередбачених ситуацій, щоб уникнути аварійного завершення програми.

### **Недостатня перевірка умов:**

Помилка: Не всі можливі випадки враховані у умовах програми.

Виправлення: Переконайтеся, що всі можливі варіанти вхідних даних розглянуті та оброблені у програмі.

## **7. Висновки**

У цій лабораторній роботі було розроблено консольну програму на мові програмування C++, яка включає дев'ять різних завдань.

Кожне завдання вимагало введення даних від користувача, обробки цих даних та виведення результату на екран. Крім того, було додано можливість вибору завдання за номером через головне меню програми.

В процесі розробки, було важливо враховувати потенційні помилки, такі як некоректне введення даних, невірна обробка винятків та несправжня обробка умов. Додатково, блок-схеми могли б бути використані для кращого розуміння логіки кожного завдання та виявлення можливих помилок.

У разі виявлення помилок, важливо було їх виправити та перевірити програму на різних вхідних даних для впевненості в її коректності.

У цілому, ця лабораторна робота надала можливість поглибити розуміння основ програмування на мові C++, вирішити різноманітні завдання та навчитися виявляти та виправляти можливі помилки в програмах.