



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Інститут прикладного системного аналізу
Кафедра системного проєтування

Алгоритми та структури даних
Лабораторна робота №2
“Дослідження базових алгоритмів”

Київ – 2020

1 Мета роботи

Ознайомитись і дослідити базові класичні алгоритми: пошук простих чисел, переведення числа з однієї системи числення в іншу, рекурсивні алгоритми. Набути навичок їх реалізації мовою програмування C/C++, навчитися вимірювати та порівнювати швидкодію алгоритмів.

2 Короткі теоретичні відомості

- Перевірка числа на простоту перебором дільників:
<https://foxford.ru/wiki/informatika/proverka-prostoty-pereborom-deliteley>
- «Решето Ератосфена»: https://uk.wikipedia.org/wiki/Решето_Ератосфена
- Переведення чисел між системами числення:
<https://coderlessons.com/tutorials/akademicheskii/uchim-kompiuternuii-logicheskuii-organizatsiiu/preobrazovanie-sistemy-schisleniia-2>
- Рекурсія: <https://ravesli.com/urok-107-rekursiya-chisla-fibonachchi-i-faktorial>
- Заміряти час виконання коду:

```
#include <time.h> // або <ctime>

clock_t start = clock();
// Тут ваш код, час виконання якого замірюється
clock_t end = clock();
float seconds = (float)(end - start) / CLOCKS_PER_SEC;
```

Якщо ваш код виконується занадто швидко, його можна занести у цикл на 1000 повторень.

- Підключити українську мову в консолі:
<https://cherto4ka.xyz/2018/10/18/українська-мова-в-консолі>

```
#include <windows.h>

void main()
{
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
}
```

Також необхідно змінити шрифт в параметрах консолі на «Lucida Console».

3 Завдання

Написати програми для виконання 3 різнотипних завдань:

- 1) Знаходження простих чисел, алгоритм «Решето Ератосфена».

В цьому завданні потрібно реалізувати два алгоритми пошуку простих чисел: звичайний перебір та «Решето Ератосфена». За допомогою обох алгоритмів розв'язати поставлену задачу та порівняти їх швидкодію, вимірявши час виконання

- 2) Переведення чисел між системами числення

- 3) Задача на рекурсивний алгоритм

Перевірити правильність роботи програм з різними вхідними даними. Зробити скріншоти результатів виводу програм.

В завданнях, де вводяться будь-які змінні, їх встановлення необхідно реалізувати через введення даних з терміналу (напр. cin).

Важливо! Дозволені бібліотеки для підключення:

- бібліотеки введення / виведення даних: `<iostream>`, `<stdio.h>`, `<conio.h>`
- бібліотека математичних функцій: `<cmath>` (`math.h`)
- бібліотеки для функції рандомайзу: `<cstdlib>` (`stdlib.h`), `<ctime>` (`time.h`)
- бібліотека для підтримки кирилиці при виведенні даних: `<locale>` (`locale.h`)
- бібліотека для підтримки українських літер в консолі Windows: `<windows.h>`

4 Зміст звіту

Звіт має містити:

- 1) Титульний аркуш
- 2) Мету роботи
- 3) Варіант завдання
- 4) Хід виконання роботи
 - a) Умова кожної задачі
 - b) Скріншот результату виконання кожного завдання
 - c) Лістинг програми (код) до кожного завдання
- 5) Висновки

5 Контрольні питання

- 1) Що впливає на різницю у часі між виконанням алгоритму знаходження простих чисел перебором дільників та алгоритму «Решето Ератосфена»?
- 2) Яка основна проблема з'являється при переведенні дійсного числа 0.3 у двійкову систему числення?
- 3) Яка принципова різниця між використанням рекурсії та циклів? В чому перевага кожного з варіантів?

6 Варіанти завдань

Варіант 1

1. Дано натуральні числа A та B , такі що $A < B$. Знайти всі прості числа на проміжку $[A, B]$, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм «Решето Ератосфена» для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 5-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано невід'ємні цілі числа A та B . Обчислити A в степені B за допомогою рекурсії (без використання циклів).

Варіант 2

1. Знайти перші 100 простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм «Решето Ератосфена» для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в

6-ву систему числення, і функцію для зворотнього переводу.

Протестувати на декількох прикладах.

3. Порахувати кількість чисел у послідовності Фібоначчі, менших заданого натурального числа N , за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 10$ програма має видати 7.

Варіант 3

1. Дано натуральні числа A та B , такі що $A < B$. Знайти суму всіх простих чисел на проміжку $[A, B]$.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 7-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.
3. Дано два цілих числа A і B . Вивести всі числа від A до B включно, в порядку зростання, якщо $A < B$, або в порядку спадання в іншому випадку, за допомогою рекурсії (без використання циклів).

Варіант 4

1. Знайти всі трицифрові прості числа, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 8-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.

3. Дано натуральне число N . Порахувати в ньому кількість входжень цифри 7 за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 717$ програма має видати 2.

Варіант 5

1. Знайти суму всіх двоцифрових простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 9-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Перевірити, чи є число N степенем двійки за допомогою рекурсії (без використання циклів та оператора піднесення до степеня).

Варіант 6

1. Знайти 28-ме, 35-те, 46-те, 114-те за рахунком просте число.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 2-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Вивести всі його цифри по одній, розділяючи їх пробілами або новими рядками, за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 627$ програма має вивести 6 2 7.

Варіант 7

1. Знайти суму 12-го, 112-го та 212-го за рахунком простих чисел.

- 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 3-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано $N > 1$ зайчиків, що сидять в ряд. У зайчиків з непарними номерами по два вуха, а у зайчиків з парними номерами - три вуха, тому що кожен з них піднімає ще й лапку. Обчислити кількість вух у ряді зайчиків за допомогою рекурсії (без використання циклів і оператора множення). Наприклад, при $N = 4$ програма має повернути 10, при $N = 7$ програма має повернути 17.

Варіант 8

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 4-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Знайти максимальну цифру цього числа за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 6.

Варіант 9

1. Дано натуральні числа A та B , такі що $A < B$. Знайти всі прості числа на проміжку $[A, B]$, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.

- 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десяткового числа в 7-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
 3. Порахувати суму перших N чисел у послідовності Фібоначчі, за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 5$ програма має видати 7.

Варіант 10

1. Знайти перші 100 простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десяткового числа в 8-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати добуток його цифр за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 12.

Варіант 11

1. Дано натуральні числа A та B , такі що $A < B$. Знайти суму всіх простих чисел на проміжку $[A, B]$.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .

- 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 9-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати в ньому кількість входжень числа 17 за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 117176171$ програма має видати 3.

Варіант 12

1. Знайти всі трицифрові прості числа, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 2-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Перевірити, чи є число N степенем трійки за допомогою рекурсії (без використання циклів та оператора піднесення до степеня).

Варіант 13

1. Знайти суму всіх двоцифрових простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 3-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.

3. Порахувати добуток чисел, починаючи від другого числа послідовності Фібоначчі до N-го числа послідовності включно, за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 6$ програма має видати 30.

Варіант 14

1. Знайти 28-ме, 35-те, 46-те, 114-те за рахунком просте число.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 4-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Вивести всі його цифри по одній, в зворотному порядку, розділяючи їх пробілами або новими рядками, за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 627$ програма має вивести 7 2 6.

Варіант 15

1. Знайти суму 12-го, 112-го та 212-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 5-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано піраміду, яка складається з блоків. Найвищий рівень піраміди складається з 1 блоку, рівень під ним - з двох, наступний - з трьох і т.д. Обчисліть кількість блоків в піраміді за заданою кількістю рівнів $N > 1$, за

допомогою рекурсії (без використання циклів і оператора множення).

Наприклад, при $N = 2$ програма має повернути 3, при $N = 7$ програма має повернути 28.

Варіант 16

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 6-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.
3. Дано невід’ємні цілі числа A та B . Обчислити A в степені B за допомогою рекурсії (без використання циклів).

Варіант 17

1. Дано натуральні числа A та B , такі що $A < B$. Знайти всі прості числа на проміжку $[A, B]$, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 9-ву систему числення, і функцію для зворотнього переводу. Протестувати на декількох прикладах.
3. Порахувати кількість чисел у послідовності Фібоначчі, менших заданого натурального числа N , за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 10$ програма має видати 7.

Варіант 18

1. Знайти перші 100 простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 2-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано два цілих числа A і B . Вивести всі числа від A до B включно, в порядку зростання, якщо $A < B$, або в порядку спадання в іншому випадку, за допомогою рекурсії (без використання циклів).

Варіант 19

1. Дано натуральні числа A та B , такі що $A < B$. Знайти суму всіх простих чисел на проміжку $[A, B]$.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 3-ву систему числення, і функцію для зворотнього переведу. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати в ньому кількість входжень цифри 7 за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 717$ програма має видати 2.

Варіант 20

1. Знайти всі трицифрові прості числа, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.

- 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N.
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 4-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
 3. Дано натуральне число N. Перевірити, чи є число N степенем двійки за допомогою рекурсії (без використання циклів та оператора піднесення до степеня).

Варіант 21

1. Знайти суму всіх двоцифрових простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число `n` на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N.
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 5-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N. Вивести всі його цифри по одній, розділяючи їх пробілами або новими рядками, за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 627$ програма має вивести 6 2 7.

Варіант 22

1. Знайти 28-ме, 35-те, 46-те, 114-те за рахунком просте число.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число `n` на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N.
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.

2. Написати функцію для переведення натурального десятичного числа в 6-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано $N > 1$ зайчиків, що сидять в ряд. У зайчиків з непарними номерами по два вуха, а у зайчиків з парними номерами - три вуха, тому що кожен з них піднімає ще й лапку. Обчислити кількість вух у ряді зайчиків за допомогою рекурсії (без використання циклів і оператора множення). Наприклад, при $N = 4$ програма має повернути 10, при $N = 7$ програма має повернути 17.

Варіант 23

1. Знайти суму 12-го, 112-го та 212-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 7-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Знайти максимальну цифру цього числа за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 6.

Варіант 24

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 8-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.

3. Порахувати суму перших N чисел у послідовності Фібоначчі, за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 5$ програма має видати 7.

Варіант 25

1. Дано натуральні числа A та B , такі що $A < B$. Знайти всі прості числа на проміжку $[A, B]$, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 3-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати добуток його цифр за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 12.

Варіант 26

1. Знайти перші 100 простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 4-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати в ньому кількість входжень числа 17 за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 117176171$ програма має видати 3.

Варіант 27

1. Дано натуральні числа A та B , такі що $A < B$. Знайти суму всіх простих чисел на проміжку $[A, B]$.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 5-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Перевірити, чи є число N степенем трійки за допомогою рекурсії (без використання циклів та оператора піднесення до степеня).

Варіант 28

1. Знайти всі трицифрові прості числа, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 6-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Порахувати добуток чисел, починаючи від другого числа послідовності Фібоначчі до N -го числа послідовності включно, за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 6$ програма має видати 30.

Варіант 29

1. Знайти суму всіх двоцифрових простих чисел.

- 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 7-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
 3. Дано натуральне число N . Вивести всі його цифри по одній, в зворотному порядку, розділяючи їх пробілами або новими рядками, за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 627$ програма має вивести 7 2 6.

Варіант 30

1. Знайти 28-ме, 35-те, 46-те, 114-те за рахунком просте число.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 8-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано піраміду, яка складається з блоків. Найвищий рівень піраміди складається з 1 блоку, рівень під ним - з двох, наступний - з трьох і т.д. Обчисліть кількість блоків в піраміді за заданою кількістю рівнів $N > 1$, за допомогою рекурсії (без використання циклів і оператора множення). Наприклад, при $N = 2$ програма має повернути 3, при $N = 7$ програма має повернути 28.

Варіант 31

1. Знайти суму 12-го, 112-го та 212-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.

- 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 9-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
 3. Дано невід’ємні цілі числа A та B . Обчислити A в степені B за допомогою рекурсії (без використання циклів).

Варіант 32

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 2-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Порахувати кількість чисел у послідовності Фібоначчі, менших заданого натурального числа N , за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 10$ програма має видати 7.

Варіант 33

1. Дано натуральні числа A та B , такі що $A < B$. Знайти всі прості числа на проміжку $[A, B]$, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .

- 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 4-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано два цілих числа A і B . Вивести всі числа від A до B включно, в порядку зростання, якщо $A < B$, або в порядку спадання в іншому випадку, за допомогою рекурсії (без використання циклів).

Варіант 34

1. Знайти перші 100 простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 5-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати в ньому кількість входжень цифри 7 за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 717$ програма має видати 2.

Варіант 35

1. Дано натуральні числа A та B , такі що $A < B$. Знайти суму всіх простих чисел на проміжку $[A, B]$.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв'язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятичного числа в 6-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.

3. Дано натуральне число N . Перевірити, чи є число N степенем двійки за допомогою рекурсії (без використання циклів та оператора піднесення до степеня).

Варіант 36

1. Знайти всі трицифрові прості числа, а також їх кількість.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 7-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Вивести всі його цифри по одній, розділяючи їх пробілами або новими рядками, за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 627$ програма має вивести 6 2 7.

Варіант 37

1. Знайти суму всіх двоцифрових простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 8-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано $N > 1$ зайчиків, що сидять в ряд. У зайчиків з непарними номерами по два вуха, а у зайчиків з парними номерами - три вуха, тому що кожен з них піднімає ще й лапку. Обчислити кількість вух у ряді зайчиків за допомогою рекурсії (без використання циклів і оператора множення). Наприклад, при $N = 4$ програма має повернути 10, при $N = 7$ програма має повернути 17.

Варіант 38

1. Знайти 28-ме, 35-те, 46-те, 114-те за рахунком просте число.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 9-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Знайти максимальну цифру цього числа за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 6.

Варіант 39

1. Знайти суму 12-го, 112-го та 212-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
 - 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десятикового числа в 2-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Порахувати суму перших N чисел у послідовності Фібоначчі, за допомогою рекурсії (без використання циклів). Перші два значення в послідовності Фібоначчі: 0, 1. Кожне наступне значення є сумою попередніх двох значень, тому послідовність приймає вигляд: 0, 1, 1, 2, 3, 5, 8, 13, 21 тощо. Наприклад, при $N = 5$ програма має видати 7.

Варіант 40

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
 - 1) Реалізувати булеву функцію `bool isPrime(int n)`, яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.

- 2) Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N .
 - 3) Розв’язати задану задачу у 2 способи: за допомогою функції `isPrime()` та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.
2. Написати функцію для переведення натурального десяткового числа в 3-ву систему числення, і функцію для зворотнього переведення. Протестувати на декількох прикладах.
3. Дано натуральне число N . Порахувати добуток його цифр за допомогою рекурсії (без використання циклів та строкових літералів). Наприклад, при $N = 126$ програма має видати 12.