

Дисципліна: “Програмування складних алгоритмів”

Лабораторна робота №2. Рекурсивні алгоритми.

Мета роботи:

Метою лабораторної роботи є набуття практичних навичок з рекурсивними функціями.



Лабораторна робота №2. Рекурсивні алгоритми.

Методичні вказівки

Лабораторна робота спирається на знаннях отриманих при вивченні наступних питань лекції:

- Поняття рекурсії.
- Поняття прямої і непрямої рекурсії.

Завдання до лабораторної роботи:

Розробити програми згідно з алгоритмом з використанням рекурсивної функції та без використання рекурсивної функції. Оцінити час виконання та складність алгоритму.



Варіанти індивідуальних завдань

Група ТР-22:

1.	$y = \sum_{t=1}^n \frac{2t}{t^2 - \sin t}$	6.	$y = \prod_{t=1}^n \frac{2t+1}{t^2}$
2.	$y = \sum_{t=1}^n \frac{\ln t}{t^3}$	7.	$y = \prod_{t=2}^n \left(\frac{3t}{t-1} - t^2 \right)$
3.	$y = \sum_{t=2}^n t^2(t-1)$	8.	$y = \prod_{t=2}^n \frac{\sqrt{t}}{t^2 - 1}$
4.	$y = \prod_{t=1}^n \frac{1}{(3t+1)^2}$	9.	$y = \sum_{t=1}^n \frac{\cos t \sqrt{1 - \cos^2 t}}{1 + \sin^2 t}$
5.	$y = \prod_{t=1}^n \sqrt{\frac{t}{1 + e^{-t}}}$	10.	$y = \sum_{t=1}^n \frac{\sin 2t}{1 + \cos t}$

11.	$y = \prod_{t=1}^n \frac{2t}{e^t - t^2}$	16.	$y = \prod_{t=1}^n \frac{\sin t}{2t-1}$
12.	$y = \sum_{t=1}^n \frac{t^2 - 2}{e^t}$	17.	$y = \prod_{t=2}^n \frac{3t-3}{3t}$
13.	$y = \sum_{t=2}^n 3t^2 \sqrt{t-1}$	18.	$y = \prod_{t=1}^n \frac{\sin 2t}{t^2 + 1}$
14.	$y = \sum_{t=1}^n \frac{\cos^2 t - \sin^2 t}{3t-1}$	19.	$y = \sum_{t=1}^n \frac{3t + 2t^2}{t^3}$
15.	$y = \prod_{t=1}^n \sqrt{\frac{1}{t + 2 \sin t}}$	20.	$y = \sum_{t=1}^n \frac{3t^2}{e^t - e^{-t}}$

21.	$y = \sum_{t=2}^n \frac{\sin^2 t}{t^3 - 1}$	26.	$y = \prod_{t=1}^n 2 \sin 2t$
22.	$y = \sum_{t=1}^n \frac{3t-2}{2t+1}$	27.	$y = \prod_{t=2}^n \frac{\sin t}{t^2 - 1}$
23.	$y = \sum_{t=1}^n \frac{t+1}{t^2 + 1}$	28.	$y = \prod_{t=1}^n \frac{\cos t}{2t + t^3}$
24.	$y = \prod_{t=1}^n \frac{\sin^2 t}{t}$	29.	$y = \sum_{t=1}^n \frac{1}{2 \sin(e^t - 1)}$
25.	$y = \prod_{t=1}^n \frac{t-1}{3t+1}$	30.	$y = \prod_{t=1}^n \frac{1}{3t^2}$

Лабораторна робота №2. Рекурсивні алгоритми.

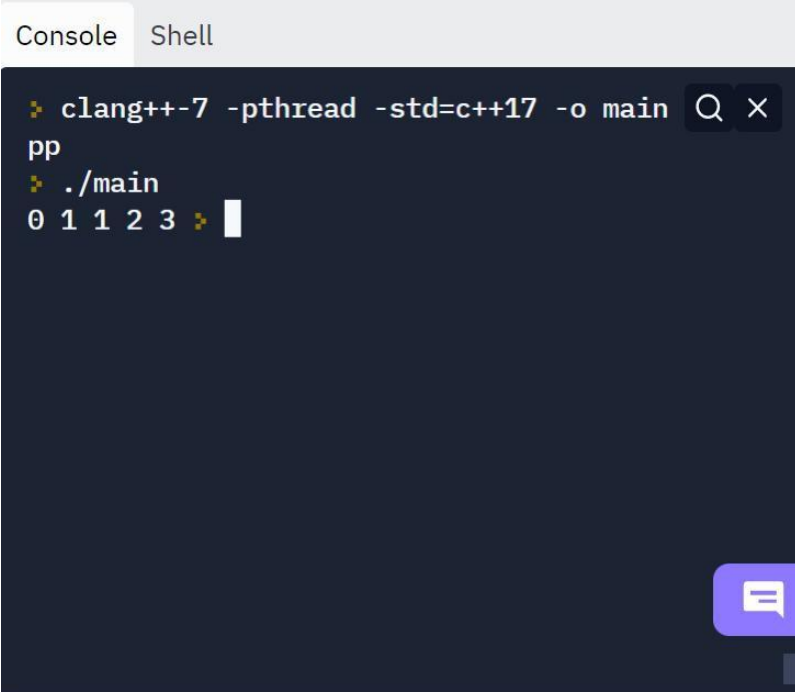
Приклад. Числа Фібоначі

$$F(n) = F(n-1) + F(n-2), \text{ де } F(0)=1, F(1)=1.$$

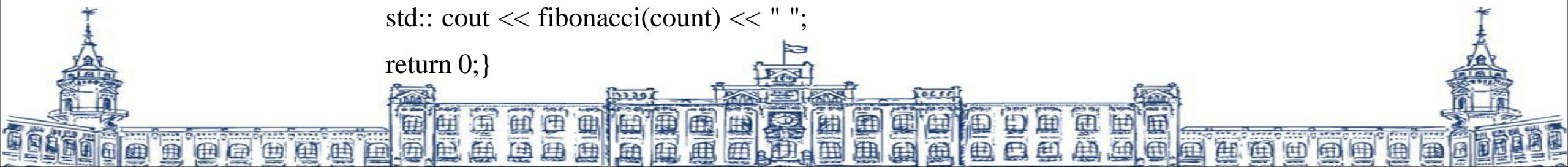
```
#include <iostream>

int fibonacci(int number)
{
    if (number == 0)
        return 0;
    else if (number == 1)
        return 1;
    return fibonacci(number-1) + fibonacci(number-2);
}

int main()
{
    for (int count=0; count < 5; ++count)
        std::cout << fibonacci(count) << " ";
    return 0;}
```



```
Console Shell
❯ clang++-7 -pthread -std=c++17 -o main pp
❯ ./main
0 1 1 2 3
```



Лабораторна робота №2. Рекурсивні алгоритми.

The screenshot shows a C++ IDE with a file named `main.cpp` and a console window. The code implements a function `sumWithoutRec` that calculates a sum using a loop and `pow` function. The console shows the compilation and execution of the program, with the input `2` and the output of the program.

```
clock_t func_start; // перевод тиків в секунды
cout << endl << "Робота функції зайняла " << timeInSeconds << " секунд" << endl << endl;
}

void sumWithoutRec(int x, int n, int N)
// Сума без рекурсії
{
    clock_t startTime = funcStartTime();
    // Початок підрахунок часу

    double sum = 0;

    for (n = 0; n <= N; n++)
    {
        sum += pow(-1, n) * x * pow(sin(x), n - 2) / pow(3, n + 1);
    }
}
```

Console output:

```
clang++-7 -pthread -std=c++17 -o main
pp
./main
Завдання №1
Ведіть значення X
>>> 2

<Оберіть варіант>
1. Діапазон n [0;10]
2. Діапазон n [0;50]
3. Перейдіть до завдання №2
>>>
```