

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

1.	$s = \prod_{x=1}^n \frac{x}{e^x - x^2}$	2.	$r = \sum_{p=3}^n \frac{\cos^2 p}{3p - 3}$
3.	$p = \prod_{a=1}^n \frac{1}{a}$	4.	$q = \sum_{l=1}^k (2l - 1)$
5.	$p = \prod_{b=1}^n \frac{\cos b}{2b - 1}$	6.	$s = \sum_{j=1}^n \frac{j^2}{e^j}$
7.	$y = \prod_{i=1}^m \frac{3i - 2}{3i}$	8.	$k = \sum_{a=1}^n \frac{a^2}{e^a - e^{-a}}$
9.	$r = \prod_{l=2}^n \frac{\sin l}{(l - 1)^2}$	10.	$u = \sum_{i=1}^m \frac{2i + 1}{i^2}$

Варіанти індивідуальних завдань

11.	$p = \prod_{k=1}^n \frac{\sin k}{k}$	12.	$s = \sum_{l=1}^p \frac{l}{l^2 + 1}$
13.	$q = \prod_{b=2}^m \frac{1}{2b - 1}$	14.	$a = \sum_{k=1}^r \frac{2k - 1}{2k + 1}$
15.	$x = \prod_{j=1}^n 2j$	16.	$f = \prod_{t=2}^n \frac{\sin^3 t}{t^2 - 1}$
17.	$p = \prod_{x=1}^n \frac{\sin x}{x^2 + 1}$	18.	$d = \prod_{k=1}^r \frac{1}{2k}$
19.	$t = \prod_{x=1}^n \frac{\cos x}{3x + x^2}$	20.	$y = \sum_{p=1}^m \frac{1}{\sin(e^p - 1)}$

Варіанти індивідуальних завдань

21.	$f = \prod_{i=1}^n \frac{2i + 1}{i^3}$	22.	$q = \sum_{p=2}^r \frac{p - 1}{p^2}$
23.	$a = \prod_{k=1}^p \left(\frac{2k}{k + 1} - k^2 \right)$	24.	$c = \sum_{j=1}^p \frac{\ln j}{j^2}$
25.	$p = \prod_{t=1}^n \frac{\sqrt{t}}{t^2 + 1}$	26.	$f = \sum_{x=1}^t \frac{2x}{x^3 - \sin x}$
27.	$s = \prod_{j=1}^n \sqrt{\frac{j^2}{1 + e^{-j}}}$	28.	$q = \sum_{i=1}^n \frac{\sin i}{1 + \cos i}$
29.	$m = \prod_{k=1}^n \frac{1}{(2k + 1)^2}$	30.	$a = \sum_{i=1}^n \frac{\sin i \cos i}{1 + \sin^2 i}$

Лабораторна робота №2. Рекурсивні алгоритми.

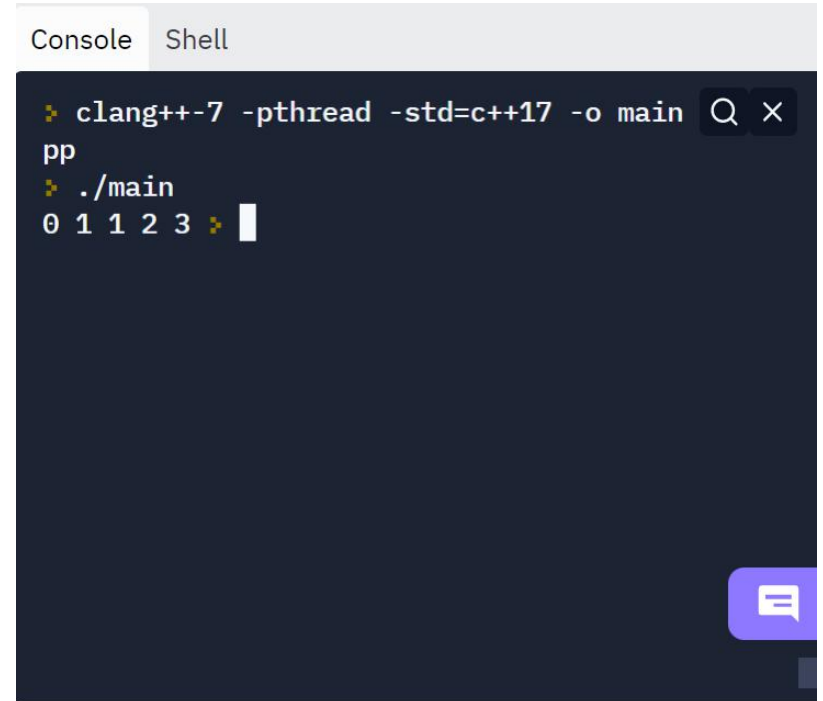
Приклад. Числа Фібоначі

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

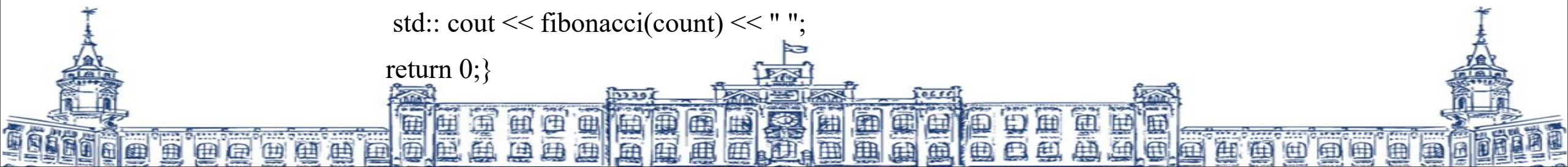
```
#include <iostream>

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
```



Дисципліна: “Програмування складних алгоритмів”

Лабораторна робота №2. Рекурсивні алгоритми.

The screenshot shows a C++ IDE with a file named `main.cpp`. The code defines a function `sumWithoutRec` that calculates a sum without using recursion. It uses `clock_t` to measure the execution time. The output is printed to the console.

```
21 cout << endl << "Робота функції  
22 зайняла " << timeInSeconds << "  
23 секунд" << endl << endl;  
24 }  
25 void sumWithoutRec(int x, int n, int N)  
26 // Сума без рекурсії  
27 {  
28 clock_t startTime = funcStartTime();  
29 // Початок підрахунок часу  
30 double sum = 0;  
31 for (n = 0; n <= N; n++)  
32 {  
33     sum += pow(-1, n) * x * pow(sin(x),  
34         n - 2) / pow(3, n + 1);  
35 }
```

The console output shows the execution time and the result of the calculation. The user has entered `2` for the variable `X`.

```
>>> 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  
>>> 
```