

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної
техніки Кафедра інформатики та програмної
інженерії

Звіт

з лабораторної роботи № 6 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження рекурсивних
алгоритмів»

Варіант 28

Виконав студент ПІ-11 Сідак Кирил Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірів Мартінова Оксана Петрівна
(прізвище, ім'я, по батькові)

Лабораторна робота №6

Дослідження рекурсивних алгоритмів

Мета – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Індивідуальне завдання:

Варіант 28

Обчислити значення функції Аккермана для двох невід’ємних цілих чисел n та m , де:

$$A(n, m) = \begin{cases} m + 1, & \text{якщо } n = 0; \\ A(n - 1, 1), & \text{якщо } n \neq 0, m = 0; \\ A(n - 1, A(n, m - 1)), & \text{якщо } n > 0, m > 0. \end{cases}$$

Постановка задачі

За допомогою підпрограми треба обчислити значення функції Аккермана відповідно до вхідних даних, тобто двох цілих невід’ємних чисел n та m , значення яких треба передати як параметри функції. Значення функції Аккермана будемо обчислювати, використовуючи рекурсію та охоронну форму оператора вибору.

Побудова математичної моделі

Складемо таблицю змінних

Змінна	Тип	Ім’я	Призначення
Перший аргумент функції Аккермана	цілий	n	Вхідне дане
Другий аргумент функції Аккермана	цілий	m	Вхідне дане
Значення, яке повертає функція	цілий	value	Проміжне дане
Значення функції Аккермана	цілий	result	Результат

Таким чином, формування задачі зводиться до обчислення значення функції Аккермана за допомогою рекурсивного алгоритму. Якщо $n == 0$ (термінальна гілка рекурсії), то значення функції – $m + 1$. Якщо перша умова не виконується (тобто $n \neq 0$) та $m == 0$, то функція викликає сама себе з аргументами $n-1$ замість n та 1 замість m . В усіх інших випадках ($n > 0$ та $m > 0$ – це усі інші випадки, адже за умовою m та n – невід’ємні цілі числа) функція викликає сама себе з аргументом $n-1$ замість n та $ackermann(n, m - 1)$ замість m , де $ackermann()$ – функція Аккермана.

Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо обчислення значення функції Аккермана.

Псевдокод

Основна програма

Початок

ввести n, m

$result = ackermann(n, m)$

вивести “Result = “, $result$

Кінець

Підпрограма

$ackermann(n, m)$

якщо $n == 0$

то

$value = m + 1$

інакше

якщо $m == 0$

то

$value = ackermann(n - 1, m)$

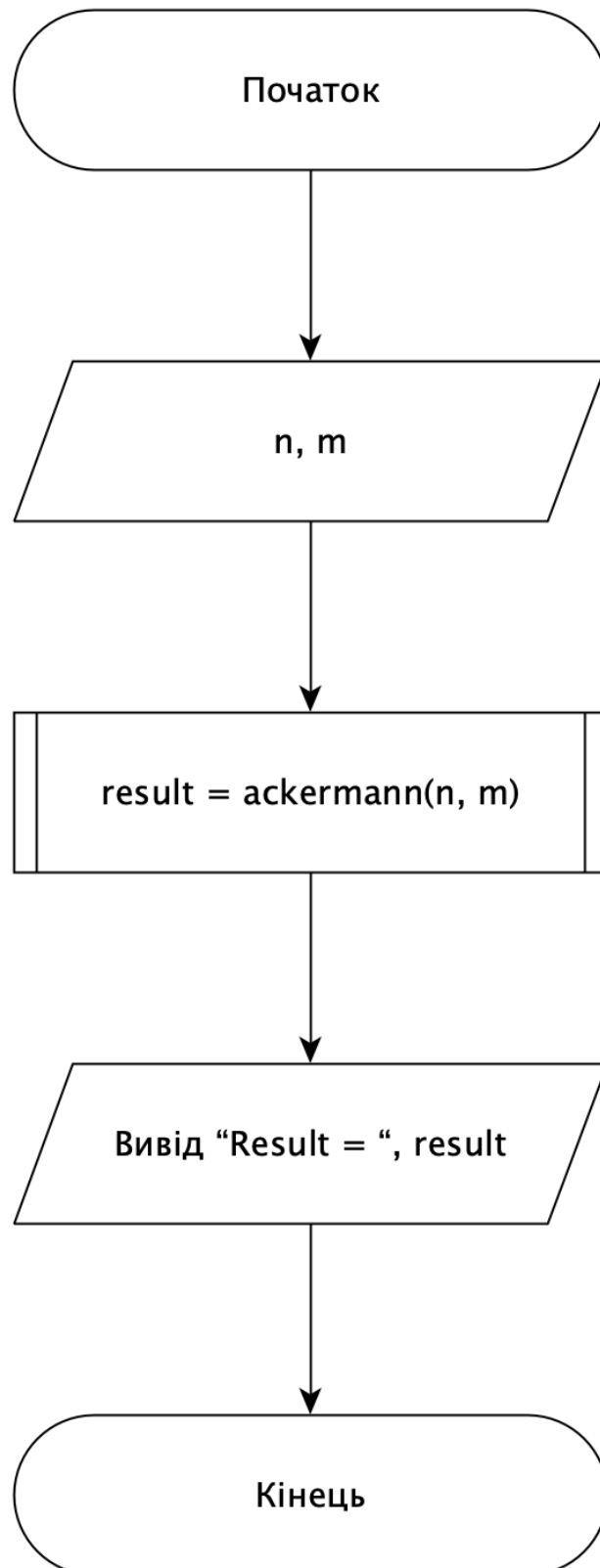
інакше

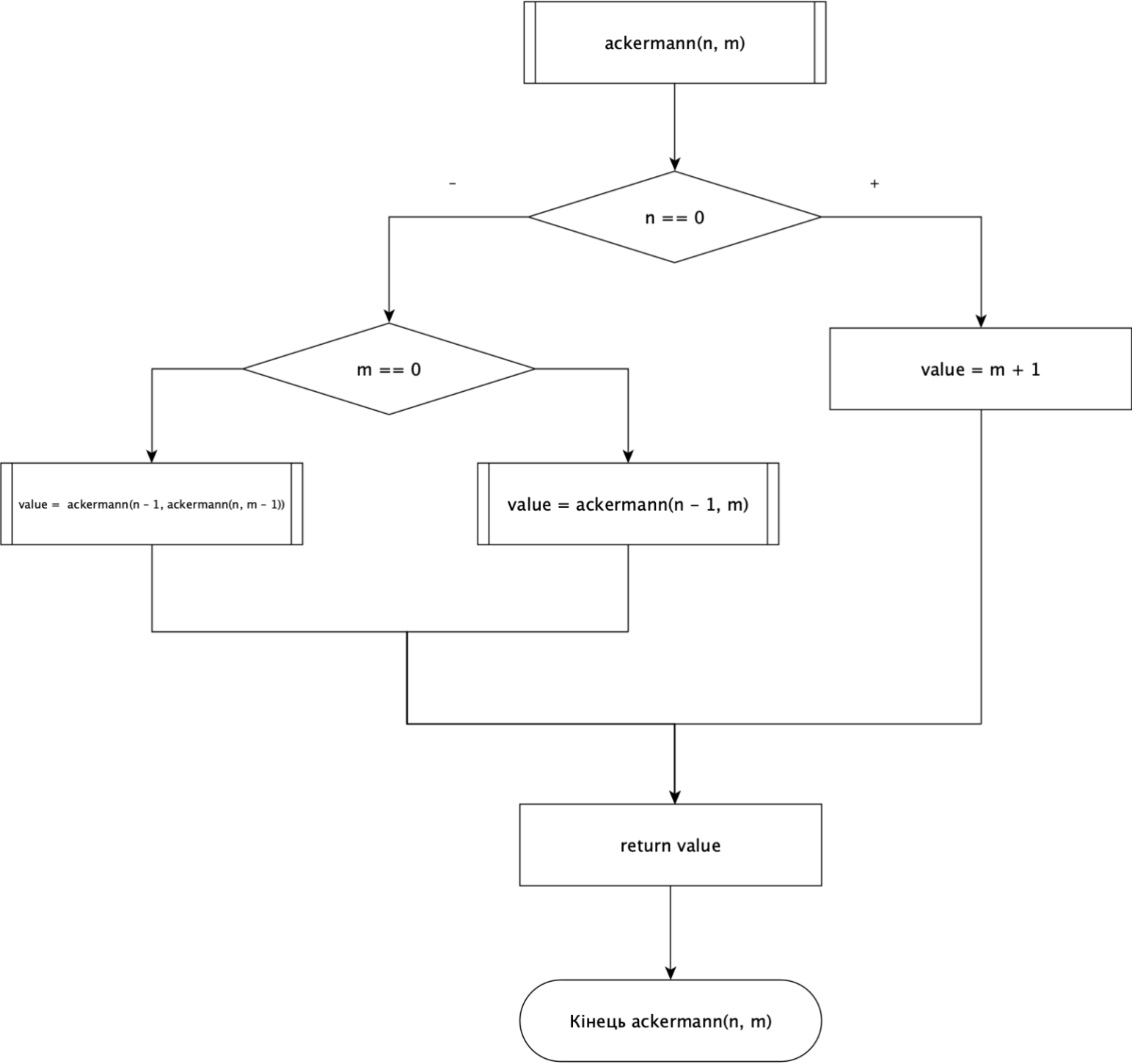
$value = ackermann(n - 1, ackermann(n, m - 1))$

все якщо

все якщо
повернути value
Кінець ackermann

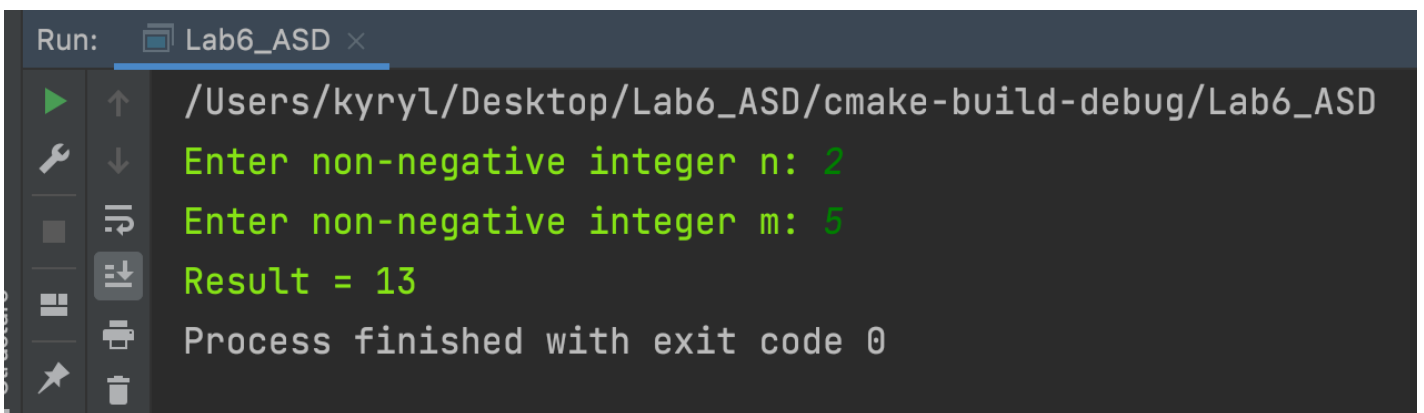
Блок-схема





Програма на C++

```
#include <iostream>
using namespace std;
int ackermann(int, int);
int main() {
    int n, m, result;
    cout << "Enter non-negative integer n: ";
    cin >> n;
    cout << "Enter non-negative integer m: ";
    cin >> m;
    result = ackermann(n, m);
    cout << "Result = " << result;
    return 0;
}
int ackermann(int n, int m) {
    int value;
    if (n == 0) {
        value = m + 1;
    }
    else {
        if (m == 0) {
            value = ackermann(n - 1, 1);
        }
        else {
            value = ackermann(n - 1, ackermann(n, m - 1));
        }
    }
    return value;
}
```



```
Run: Lab6_ASD x
/Users/kyryl/Desktop/Lab6_ASD/cmake-build-debug/Lab6_ASD
Enter non-negative integer n: 2
Enter non-negative integer m: 5
Result = 13
Process finished with exit code 0
```

Випробування алгоритму

Блок	Дія
	Початок
1	$n = 2, m = 1$
2	<code>result = ackerman(2, 5)</code>
3	<code>result = 13</code>
4	Виведення «Result = 13»
	Кінець

Висновок

Отже, я дослідив роботу рекурсивних алгоритмів, зробивши рекурсивний алгоритм для обчислення значення функції Аккермана в залежності від значень двох її аргументів шляхом розгалуження алгоритму на рекурсивну та термінальну гілки.