

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 No 6 з дисципліни  
«Основи програмування 1.  
Базові конструкції»  
«Організація підпрограм»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище,  
ім'я, по батькові)

Перевірів \_\_\_\_\_ ( прізвище, ім'я, по  
батькові)

Київ 2021

## Лабораторна робота No 6

**Тема:** Організація підпрограм.

**Мета:** Набути навичок складання і використання підпрограм користувача.

### Хід роботи

#### Задача

7. Задані невід'ємні цілі числа  $n$  і  $m$ . Обчислити функцію Акермана, користуючись наступними рекурентними співвідношеннями:

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

#### Постановка задачі

Функція Акермана приймає два невід'ємних цілих числа як параметри і повертає натуральне число. Ця функція росте дуже швидко, наприклад, число  $A(4, 4)$  настільки велике, що кількість цифр у запису цього числа багаторазово перевищує кількість атомів у видимій частині всесвіту.

Values of $A(m, n)$						
$m \backslash n$	0	1	2	3	4	$n$
0	1	2	3	4	5	$n + 1$
1	2	3	4	5	6	$n + 2 = 2 + (n + 3) - 3$
2	3	5	7	9	11	$2n + 3 = 2 \cdot (n + 3) - 3$
3	5	13	29	61	125	$2^{(n+3)} - 3$

#### Випробування коду на C++

```
#include <iostream>
using namespace std;

int akk(int m, int n){
    if(m==0){
        return n+1;
    }else if(m>0&& n==0){
        return akk(m-1, 1);
    }else if(m>0&& n>0){
        return akk(m-1, akk(m, n-1));
    }else{
        return 0;
    }
}

int main() {
    int m, n;
    cout<<"Enter m";
    cin>>m;
    cout<<"Enter n";
    cin>>n;
    return akk(m,n);
}
```

*Результат*

```
Enter m3
Enter n2
Program ended with exit code: 29
```

**Випробування коду на Python**

```
def akk(m, n):
    if m==0:
        return n+1
    elif m>0 and n==0:
        return akk(m-1, 1)
    elif m>0 and n>0:
        return akk(m-1, akk(m, n-1))
    else:
        return 0
m = int(input("Enter m"))
n = int(input("Enter n"))
print(akk(m,n))
```

*Результат*

```
Enter m2
Enter n4
11
```

**Висновок:** Для виконання завдання було написано функцію Акермана у вигляді підпрограми. При великих значеннях  $n$  та  $m$  може виникнути переповнення стеку, тому що функція Аккермана є двічі рекурсивною, один з аргументів функції є таж рекурсивна функція.