

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 22

Виконав студент _____ Мєшков Андрій Ігорович _____
(шифр, прізвище, ім'я, по батькові)

Перевірив _____ Вєчерковська Анастасія Сергіївна _____
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

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

Варіант 22

Задача. Обчислити кількість комбінацій з n різних елементів по m . Кількість комбінацій визначається формулою

$$C_n^m = \begin{cases} 1, & \text{якщо } m = 0, n > 0 \text{ або } m = n \geq 0; \\ 0, & \text{якщо } m > n \geq 0; \\ C_{n-1}^{m-1} + C_{n-1}^m & \text{в інших випадках.} \end{cases}$$

Постанова задачі. З клавіатури вводиться два числа. Числа перевіряються на невід'ємне значення та результат реалізований рекурентно за допомогою підпрограми у вигляді функцій для обчислення кількості комбінацій.

Побудова математичної моделі: для більшої наочності складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Кількість елементів більшої групи	Цілий/Невід'ємний	n	Початкові дані
Кількість елементів меншої групи	Цілий/Невід'ємний	m	Початкові дані
Перший параметр першої функції	Цілий/Невід'ємний	n_0	Проміжні дані
Другий параметр першої функції	Цілий/Невід'ємний	m_0	Проміжні дані
Результат першої функції	Цілий/Невід'ємний	c	Проміжні дані
Кількість комбінацій з різних елементів по m	Цілий/Натуральний	C_{nm}	Результат

Числа перевіряються на невід'ємність – кількість елементів не може бути менше за нуль.

Якщо $n < 0$ або $m < 0$ на екран виводиться текст: «Числа повинні бути додатніми». Якщо числа невід'ємні, то $C(n,m) = C(n-1,m) + C(n,m-1)$, де $C(n,0) = 1$ – рекурсивна функція яка обчислюється за формулою, яка складається з випадків:

1. Якщо $m=0$, $n>0$ або $n=m=0$, то $c=1$;
 2. Якщо $m>n>0$, то $c=0$;
 3. У інших випадках – $c=C(n-1,m)+C(n,m-1)$;
- c – результат, який ми повертаємо.

Тобто функція буде викликати саму себе до тих пір поки значення параметрів не будуть задовольняти перший випадок. Результат – це сума великої кількості одиниць. Кінцевий результат виводиться на екран.

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію перевірки чисел на від'ємність та обчислення кількості комбінацій.

Псевдокод **Програма**

Крок 1
Початок

введення n, m
Перевірка чисел на
від'ємність та
обчислення кількості
комбінацій

Кінець

Крок 2
Початок

введення n, m
якщо n<0 || m<0
то
виведення: «Числа повинні бути
додатними»
інакше
 $C_{nm} := C(n, m)$
все якщо
Кінець

Підпрограма

C(n0, m0):

C(n0, m0)

якщо ($m0 == 0 \&\& n0 > 0$) || ($m0 == n0 \&\& n0 >= 0$)

то

c:=1

інакше якщо $m0 > n0 \&\& n0 >= 0$

то

c:=0

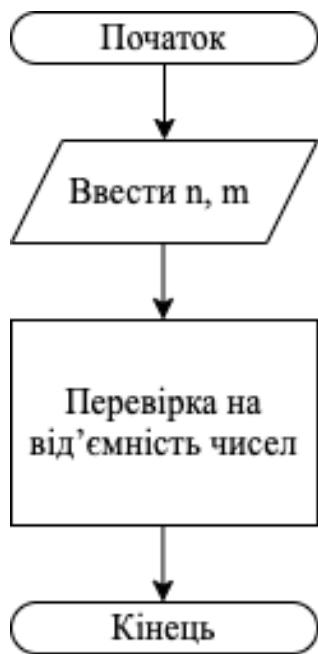
інакше

c:=C(n0-1, m0-1) + C(n0-1, m0)

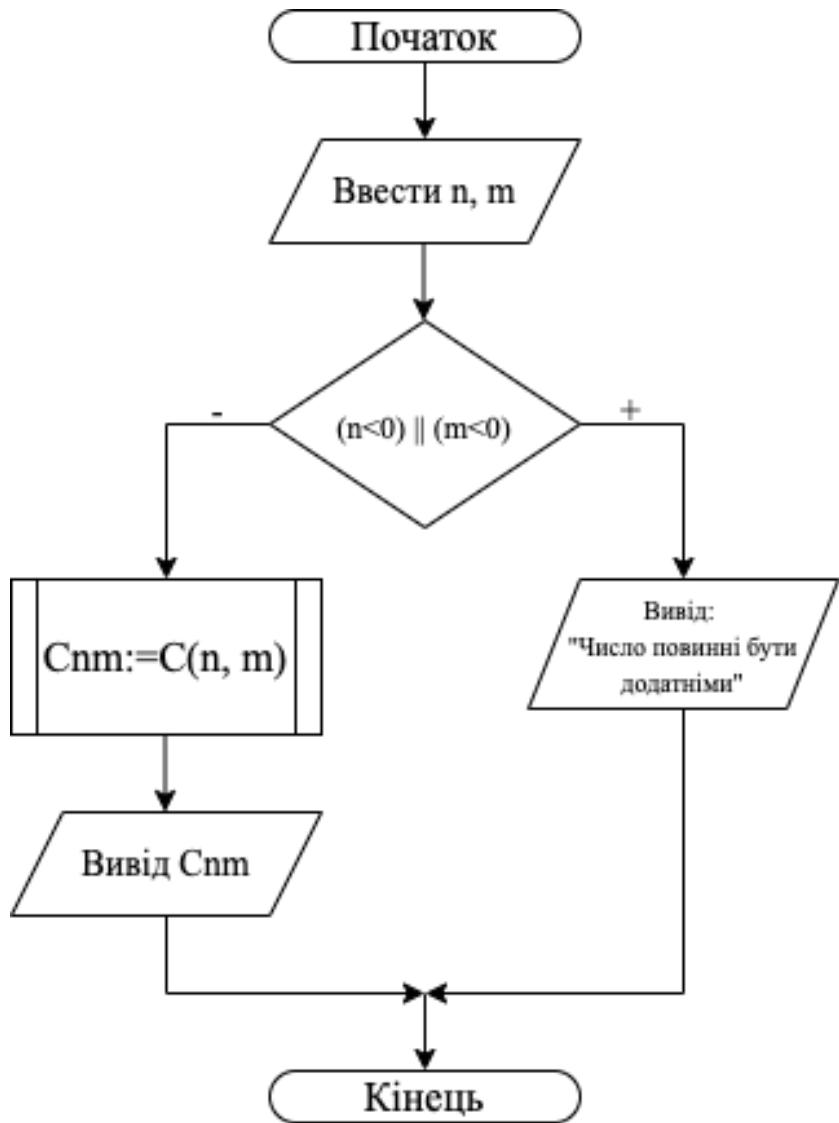
повернути c

Блок-схема Програма:

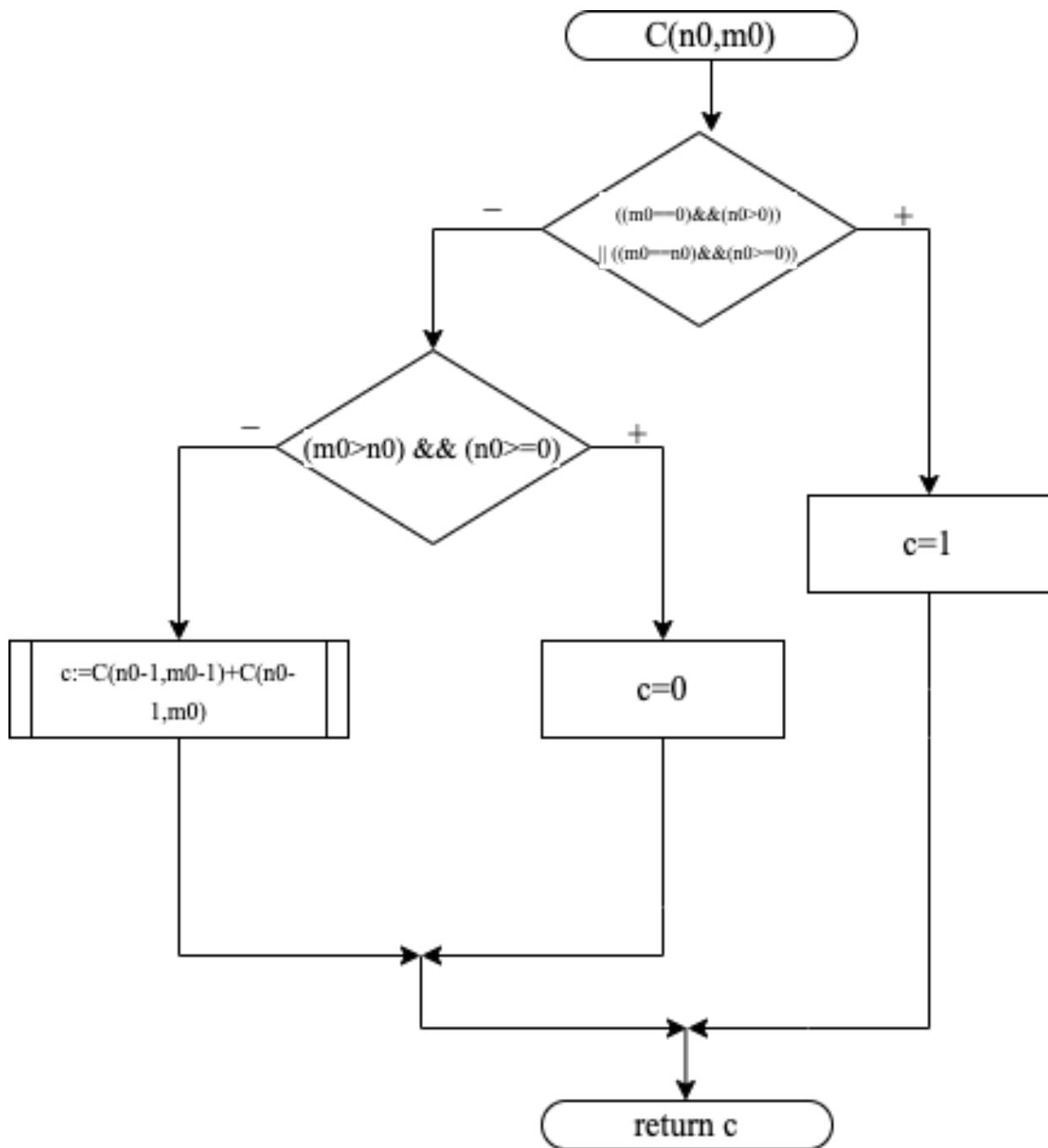
Крок 1



Крок 2



Підпрограми:



Випробування алгоритму: перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних.

Код програми:

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 long long C(int n0, int m0){
6     long long c;
7     if(((m0==0)&&(n0>0))||( (m0==n0)&&(n0>=0))) {
8         c = 1;
9     }
10
11    else if((m0>n0)&&(n0>=0)){
12        c = 0;
13    }
14
15    else{
16        c = C(n0-1,m0-1) + C(n0-1,m0);
17    }
18    return c;
19 }
20 int main ()
21 {
22     int n, m;
23     long long Cnm;
24     cout << "n = ";
25     cin >> n;
26     cout << "m = ";
27     cin >> m;
28
29     if(n<0 | | m<0){
30         cout << "n and m must be positive." << endl;
31     }
32
33     else{
34         Cnm=C(n, m);
35         cout << "Cnm = " << Cnm << endl;
36     }
37 }
38 |
```

Tect№1(10, 5)

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 long long C(int n0, int m0){
6     long long c;
7     if(((m0==0)&&(n0>0))||( (m0==n0)&&(n0>=0))) {
8         c = 1;
9     }
10
11    else if((m0>n0)&&(n0>=0)){
12        c = 0;
13    }
14
15    else{
16        c = C(n0-1,m0-1) + C(n0-1,m0);
17    }
18    return c;
```

```
n = 10
m = 5
Cnm = 252
Program ended with exit code: 0|
```

Tect№2(-5, 0)

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 long long C(int n0, int m0){
6     long long c;
7     if(((m0==0)&&(n0>0))||( (m0==n0)&&(n0>=0))) {
8         c = 1;
9     }
10
11    else if((m0>n0)&&(n0>=0)){
12        c = 0;
13    }
14
15    else{
16        c = C(n0-1,m0-1) + C(n0-1,m0);
17    }
18    return c;
```

```
n = -5
m = 0
n and m must be positive.
Program ended with exit code: 0|
```

Тест№3(23, 233)

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 long long C(int n0, int m0){
6     long long c;
7     if(((m0==0)&&(n0>0))||( (m0==n0)&&(n0>=0))) {
8         c = 1;
9     }
10
11    else if((m0>n0)&&(n0>=0)){
12        c = 0;
13    }
14
15    else{
16        c = C(n0-1,m0-1) + C(n0-1,m0);
17    }
18    return c;
```

n = 23
m = 233
Cnm = 0
Program ended with exit code: 0

Висновок: отже, за допомогою підпрограми(рекурсивна функція) було організовано знаходження кількості комбінацій з n різних елементів по m. Було досліджено рекурсивні алгоритми, проаналізовано подане завдання, декомпозовано та виконано. Також були розроблені псевдокод, код програми та блок-схема поставленого алгоритму.