

### Завдання

$$\sum_{k=1}^n \frac{a_k - b_k}{k!},$$
$$a_1 = 1, \quad a_k = 0.5 \left( \sqrt{b_{k-1}} + 5\sqrt{a_{k-1}} \right),$$
$$b_1 = 1, \quad b_k = 2a_{k-1}^2 + b_{k-1}.$$

### Постановка задачі

Необхідно за допомогою рекурсивних підпрограм по обчисленню  $a$ ,  $b$ ,  $k!$  накопичити вказану суму. Результатом виконання алгоритму є виведення обчисленої суми, вхідних даних достатньо.

### Розв'язання

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

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо обчислення суми

Крок 3. Деталізуємо обчислення  $a$

Крок 4. Деталізуємо обчислення  $b$

Крок 5. Деталізуємо обчислення факторіалу

### Псевдокод

#### Основна програма

**Start**

**input**  $n$

$sum = \text{FindSum}(n)$

**output**  $sum$

**End**

### Підпрограми

**FindSum( $n$ )**

$sum = 0$

**repeat for**  $k$  **from** 1 **to**  $n$

sum += (FindA(k - 1) - FindB(k - 1))/ FindFactorial(k)

**end repeat**

**End FindSum**

**FindA(index)**

**if** index == 0

**then**

result = 1

**else**

result = 0.5 \* (sqrt(FindB(index - 1)) + 5 \* sqrt(FindA(index - 1)))

**end if**

**return** result

**End FindA**

**FindB(index)**

**if** index == 0

**then**

result = 1

**else**

result = 2 \* FindA(index - 1) ^ 2 + FindB(index - 1)

**end if**

**return** result

**End FindB**

**FindFactorial(index)**

**if** index == 0 **or** index == 0

**then**

result = 1

else

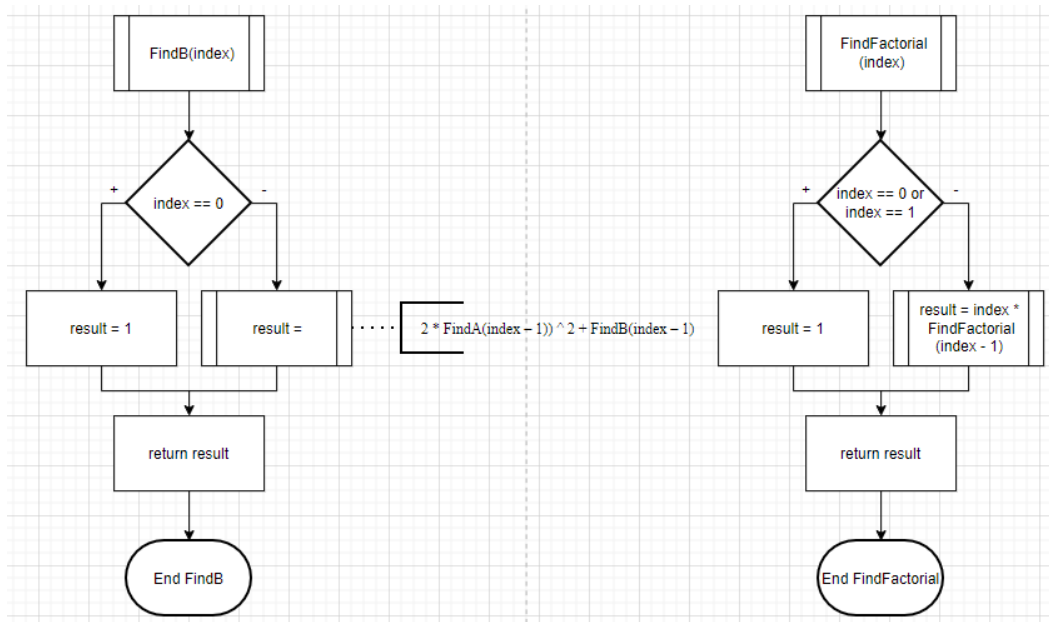
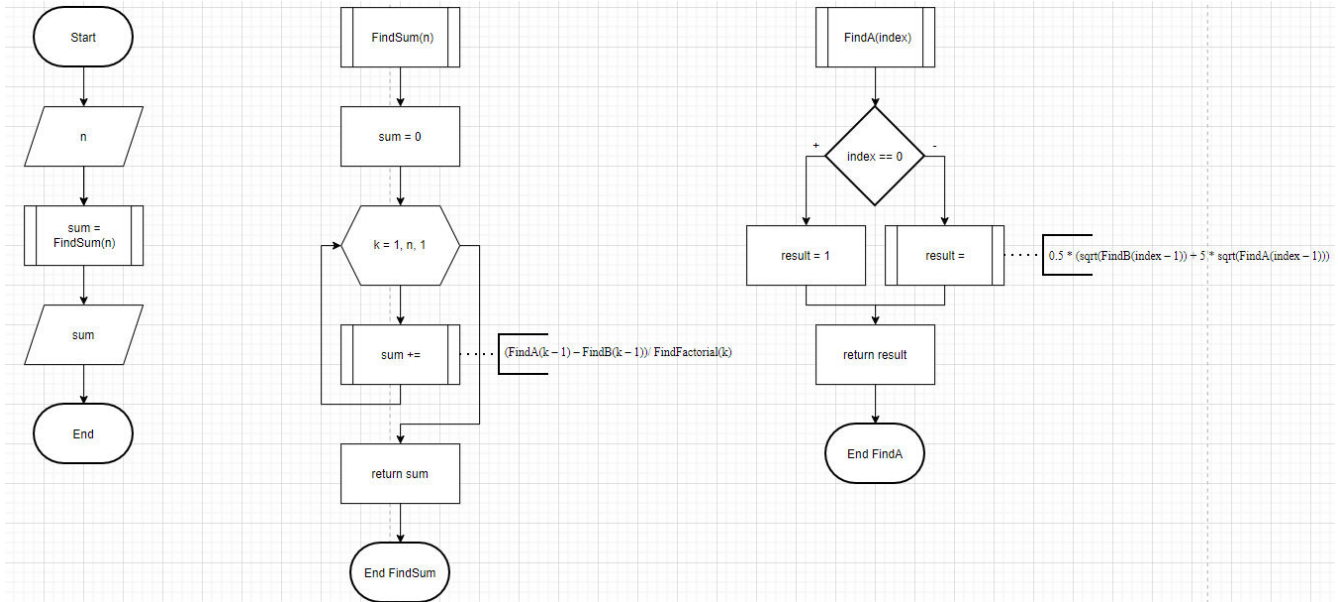
result = index \* FindFactorial(index - 1)

end if

return result

End FindFactorial

## Блок-схема



## Тестування алгоритму

Блок	Дія
	Початок
1	$n = 3$
2	Після першої ітерації в FindSum: FindA(0) = 1 FindB(0) = 1 FindFactorial(1) = 1 sum = 0
3	Після другої ітерації в FindSum: FindA(1) = 3 FindB(1) = 3 FindFactorial(2) = 2 sum = 0
4	Після третьої ітерації в FindSum: FindA(2) = 5.2 FindB(2) = 21 FindFactorial(3) = 6 sum = -2.63
5	Виведення -2.63
	Кінець

## Код програми

```
#include <iostream>
```

```
double FindSum(int n);  
double FindA(int index);  
double FindB(int index);  
double FindFactorial(int index);
```

```
int main()  
{  
    int n;  
    double sum = 0;  
  
    std::cout << "Enter n(positive whole number): ";  
    std::cin >> n;  
  
    sum = FindSum(n);  
  
    std::cout << "\nThe sum is: " << sum;  
}
```

```
double FindSum(int n)  
{  
    double sum = 0;  
    for (int k = 0; k < n; k++)  
    {
```

```

        sum += (FindA(k) - FindB(k)) / FindFactorial(k + 1);
    }
    return sum;
}

double FindA(int index)
{
    double result;
    if (index == 0)
    {
        result = 1;
    }
    else
    {
        result = 0.5 * (sqrt(FindB(index - 1)) + 5 * sqrt(FindA(index - 1)));
    }
    return result;
}

double FindB(int index)
{
    double result;
    if (index == 0)
    {
        result = 1;
    }
    else
    {
        result = 2 * FindA(index - 1) * FindA(index - 1) + FindB(index - 1);
    }
    return result;
}

double FindFactorial(int index)
{
    double result;
    if (index == 0 || index == 1)
    {
        result = 1;
    }
    else
    {
        result = index * FindFactorial(index - 1);
    }
    return result;
}

```

```

Enter n(positive whole number): 3
The sum is: -2.63397

```