

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

**Лабораторна робота №2**  
з дисципліни  
«Алгоритми і структури даних»

Виконала:  
Студентка групи ІМ-12  
Миць Вікторія Ігорівна  
Номер у списку групи: 19

Перевірила:  
Молчанова А. А

Київ 2021

### Завдання:

Задано натуральне число  $n$ . Вирахувати значення заданої формули за варіантом.

Для вирішення задачі написати дві програми:

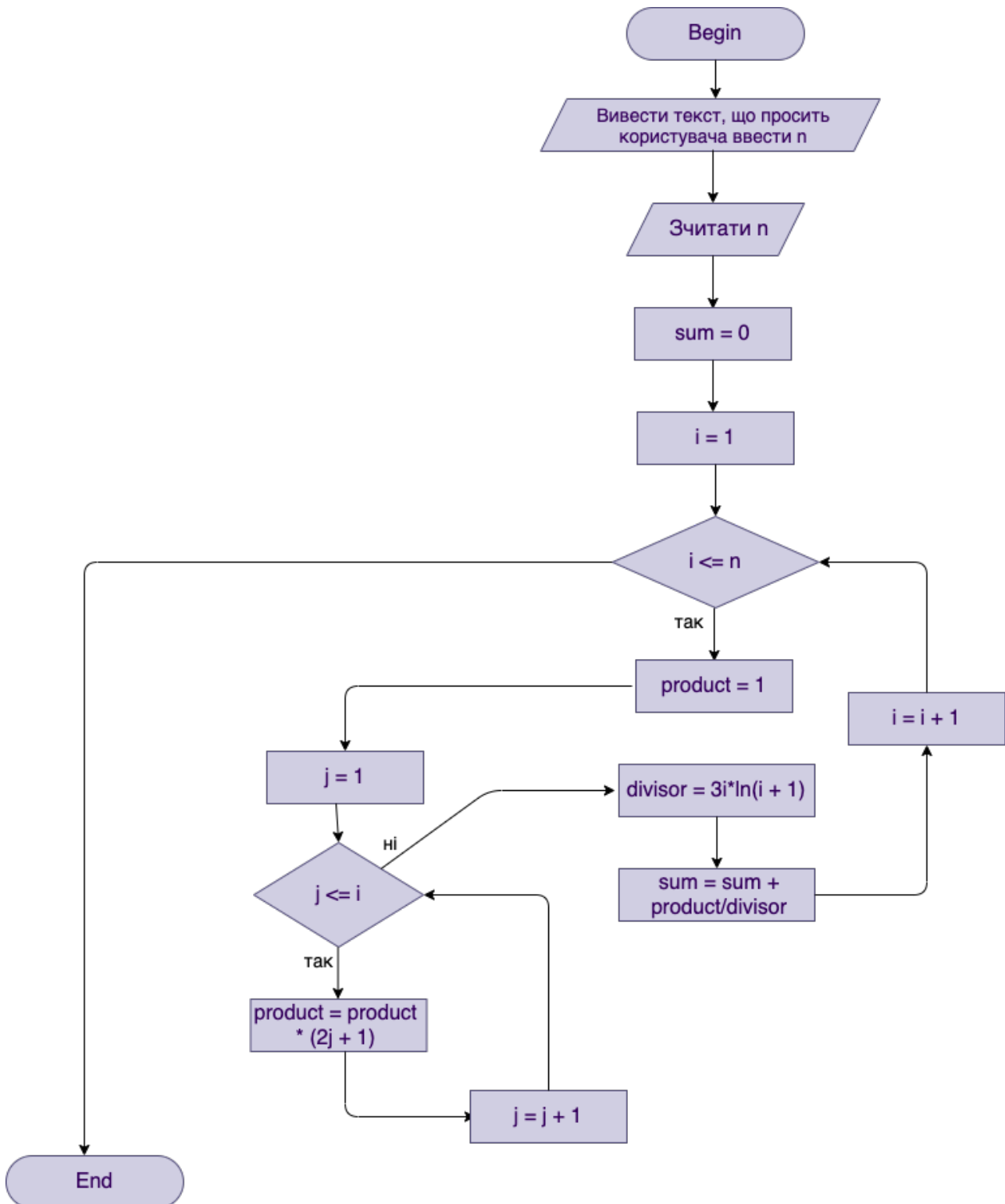
- 1) Перша програма повинна використовувати для обчислення формули вкладені цикли
- 2) Друга програма повинна виконати обчислення формули за допомогою одного циклу з використанням методу динамічного програмування

### Варіант 19:

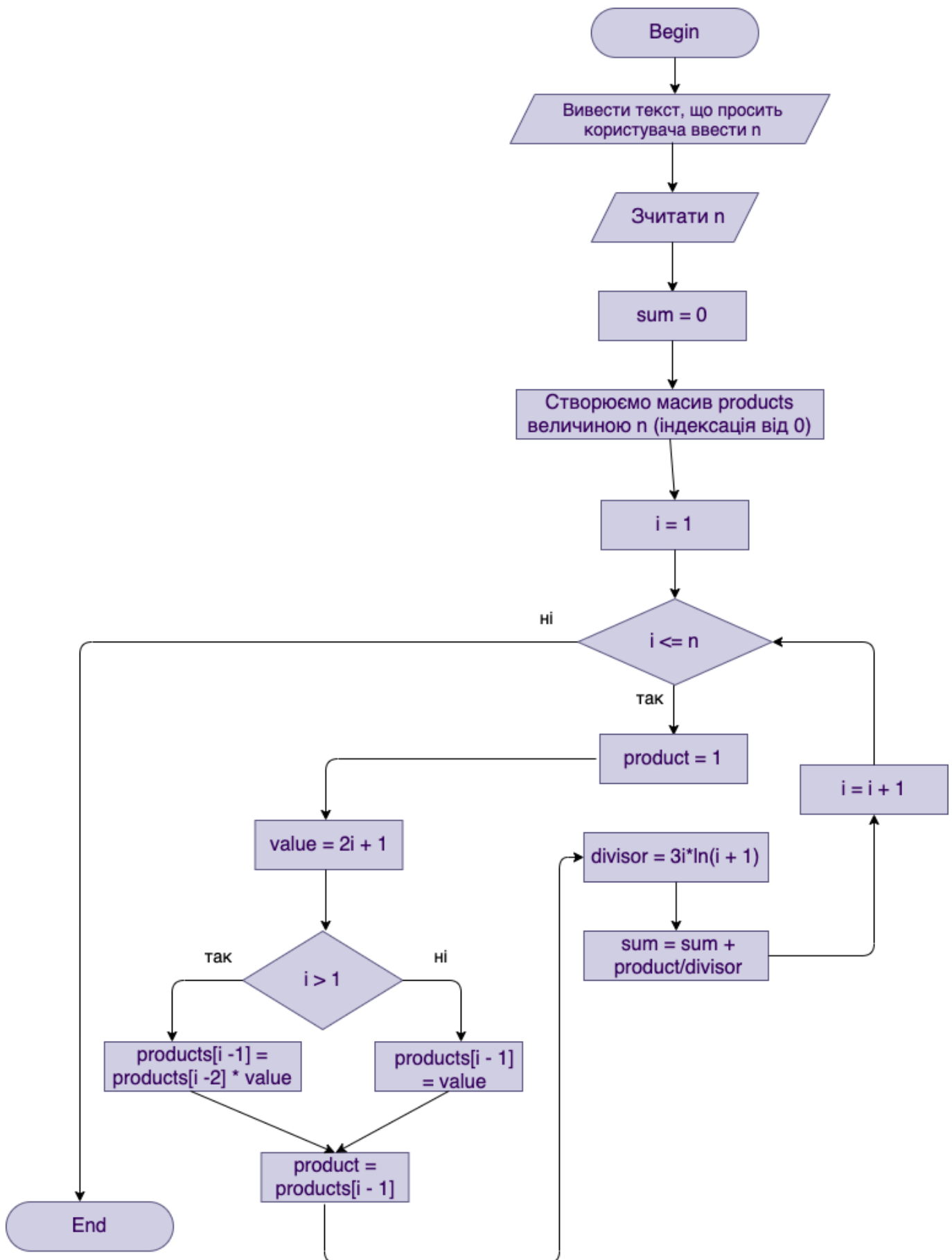
19.	$S = \sum_{i=1}^n \frac{\prod_{j=1}^i (2j + 1)}{3i \cdot \ln(i + 1)}$
-----	-----------------------------------------------------------------------

## Діаграми алгоритмів

1) Із вкладеним циклом



2) За допомогою динамічного програмування



Код алгоритму на C:

1)

```
#include <stdio.h>
#include <math.h>

int operation_count = 0; //counting only arithmetic operations

double evaluateProduct(int i) {
    double product = 1;
    for (int j = 1; j <= i; j++) {
        product *= 2*j + 1;
        operation_count++;
    }
    return product;
}

double evaluateResult(int n) {
    double sum = 0;
    for (int i = 1; i <= n; i++) {
        double product = evaluateProduct(i);
        double divisor = 3*i*log(i + 1);
        sum += product/divisor;
        operation_count += 2; //adding two because the operations in
        evaluateProduct are already counted
    }
    return sum;
}

int main() {
    int n;
    printf("Input N value: ");
    scanf("%d", &n);

    double result = evaluateResult(n);
    printf("Result: %lf", result);
    printf("\n");
    printf("Count of perfomed operations: %d", operation_count);

    return 0;
}
```

2)

```
#include <stdio.h>
#include <math.h>

int operation_count = 0; //counting only arithmetic operations

double evaluateProduct(int i, double products[]) {
    int value = 2*i + 1;
    if (i > 1) {
        products[i - 1] = products[i - 2] * value;
    } else {
        products[i - 1] = value;
    }
    return products[i - 1];
}

double evaluateResult(int n) {
    double sum = 0;
    double products[n];
    for (int i = 1; i <= n; i++) {
        double product = evaluateProduct(i, products);
        double divisor = 3*i*log(i + 1);
        sum += product/divisor;
        operation_count += 3; //adding three because there can be only one
        arithmetic operation in evaluateProduct
    }
    return sum;
}

int main() {
    int n;
    printf("Input N value: ");
    scanf("%d", &n);

    double result = evaluateResult(n);
    printf("Result: %lf", result);
    printf("\n");
    printf("Count of perfomed operations: %d", operation_count);

    return 0;
}
```

## Результати тестувань і обчислення на калькуляторі

```
vika@MacBook-Pro-vika 1.2 Цикли % ./code
Input N value: 1
Result: 1.442695%
```

$$\sum_{n=1}^k \left( \frac{\prod_{j=1}^n (2j+1)}{3n \cdot \ln(n+1)} \right)$$

= 1.44269504089

$k = 1$

\_\_\_\_\_  $\leq k \leq$  \_\_\_\_\_ Step: \_\_\_\_\_

```
vika@MacBook-Pro-vika 1.2 Цикли % ./code
Input N value: 2
Result: 3.718293%
```

$$\sum_{n=1}^k \left( \frac{\prod_{j=1}^n (2j+1)}{3n \cdot \ln(n+1)} \right)$$

= 3.71829310746

$k = 2$

\_\_\_\_\_  $\leq k \leq$  \_\_\_\_\_ Step: \_\_\_\_\_

```
vika@MacBook-Pro-vika 1.2 Цикли % ./code
Input N value: 3
Result: 12.134014%
```

$$\sum_{n=1}^k \left( \frac{\prod_{j=1}^n (2j+1)}{3n \cdot \ln(n+1)} \right)$$

= 12.1340141793

$k = 3$

\_\_\_\_\_  $\leq k \leq$  \_\_\_\_\_ Step: \_\_\_\_\_

```
vika@MacBook-Pro-vika 1.2 Цикли % ./code
Input N value: 5
Result: 447.83480476
```

$$\sum_{n=1}^k \left( \frac{\prod_{j=1}^n (2j+1)}{3n \cdot \ln(n+1)} \right)$$

= 447.834804476

k = 5

\_\_\_\_\_ ≤ k ≤ \_\_\_\_\_ Step: \_\_\_\_\_

Таблиця з кількістю операцій

n		1	2	3	5	10	20	30	50	100
Кількість операцій	1 спосіб	3	7	12	25	75	250	525	1375	5250
	2 спосіб	3	6	9	15	30	60	90	150	300