

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 № 3 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Лінійні алгоритми»

Варіант 1

Виконав студент ІП-45 Янов Богдан Євгенійович

Перевірила старший викладач Вечерковська Анастасія Сергіївна

Київ 2024

Лабораторна робота №3

Тема - циклічні алгоритми.

Мета – дослідити подання операторів повторення дій та набути практичних навичок їх використання під час складання циклічних програмних специфікацій.

Завдання

Скласти програму для обчислення значення елементарної функції в заданій точці за допомогою степеневого ряду з точністю до члена, меншого за 0.00001. Визначити кількість доданків.

Порівняти отримане значення суми зі значенням вказаної функції, обчисленим за допомогою бібліотечних функцій.

$$SH = \sum_{k=0}^{\infty} \frac{x^{2k+1}}{(2k+1)!}$$

Постановка задач

Якщо потрібно обчислити значення елементарної функції в заданій точці за допомогою степеневого ряду з точністю до члена, меншого за 0.00001 (передбачається, що точне значення обчислити неможливо) та визначити кількість доданків, потрібно використовувати наступний алгоритм:

1. Обчислити за формулою $x^{(2k+1)} / (2k+1)!$ значення term.
2. Додати term до sh.
3. Збільшити k, тобто кількість доданків, на 1.
4. Перевірити істинність рівності $|term| \geq 0.00001$. Якщо вона є істинною, перейти до кроку 1, інакше до кроку 5.
5. Записати значення sh, k.

Оскільки кількість повторень невідома, будемо використовувати ітераційний цикл. Програмну специфікацію напишемо в псевдокоді та графічній формі у вигляді блок-схеми.

Математична модель

Name	Type	Variable	Purpose
x	Real	Series parameter	Initial data
sh	Real	Sum of the series	Result
k	Integer	Starting value from which the sum of the series begins	Additional data
term	Real	Current value	Additional data

Псевдокод

Step 1

Begin

Input x

Initialization of sh, k, term

Calculation of sh

Output sh, k

Comparison of sh

End

Step 2

Begin

Input x

sh := 0

k := 0

term := 0

Calculation of sh

Output sh, k

Comparison of sh

End

Step 3

Begin

Input x

sh := 0

k := 0

term := 0

Do

term = $x^{(2k+1)} / (2k+1)!$

sh += term

k += 1

While |term| >= 0.00001

Output sh, k

Comparison of sh

End

Step 4

Begin

Input x

sh := 0

k := 0

term := 0

Do

term = $x^{(2k+1)} / (2k+1)!$

sh += term

k += 1

While |term| >= 0.00001

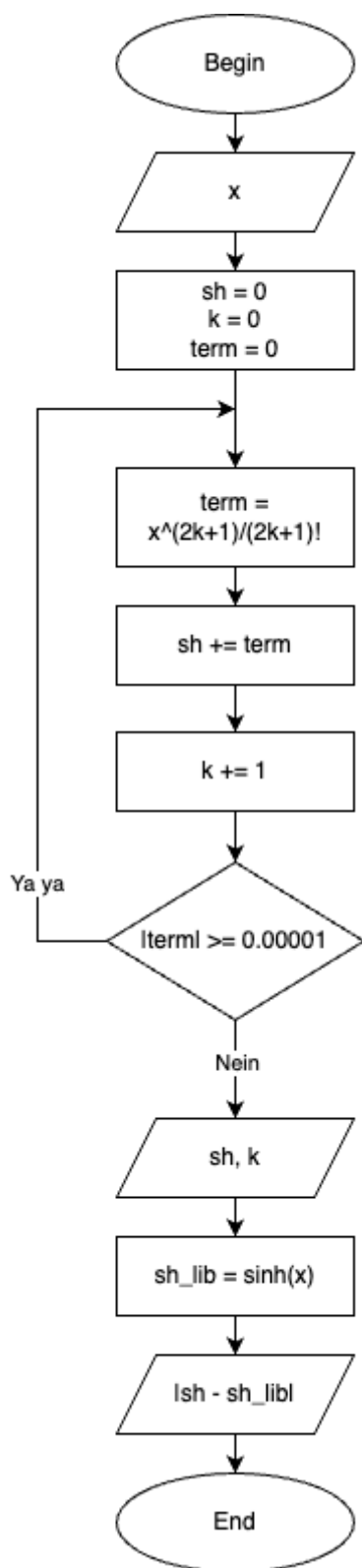
Output sh, k

sh_lib := sinh(x)

Output |sh - sh_lib|

End

Блок-схема (крок 4)



Тестування алгоритму

1. $x = 0.1$
2. $sum = 0$
3. $term = 0.1^1 / 1 = 0.1$
4. $sum += term$
5. $|term| \geq 0.00001$
6. $term = 0.1^3 / 6 = 0,10016675$
7. $sum += term$
8. $|term| \geq 0.00001$
9. $term = 0.1^5 / 120 = 8,333e-8$
10. $sum += term$
11. $|term| < 0.00001$
12. $sh_lib = \sinh(x)$
13. $sum \approx 0,10016675, sh_lib = 0,100167$

Код мовою C

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

double fact(int n) {
    if (n == 0 || n == 1)
        return 1;
    return n * fact(n - 1);
}

int main() {
    double x;
    double sh = 0;
    int k = 0;
    double term;

    scanf("%lf", &x);

    do {
        term = pow(x, 2 * k + 1) / fact(2 * k + 1);
```

```
        sh += term;

        k++;
    } while (fabs(term) >= 0.00001);

    printf("%lf %i\n", sh, k);

    double sh_lib = sinh(x);
    printf("%lf\n", fabs(sh - sh_lib));

    return 0;
}
```

Висновок: Дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій.