

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК

Лабораторна робота № _ 2 _

З дисципліни:
«Вступ до спеціальності»

Виконав: Тахтай Н.В. група № КН-42/2
Перевірила: доцент, к.т.н. Москаленко А.С.

СУМИ – 2024

Лабораторна робота 2.

Тема : Запис алгоритмів за допомогою псевдокоду

Мета :

1. Зрозуміти концепцію псевдокоду та його використання в процесі розробки алгоритмів.
2. Розвинути навички запису алгоритмів використовуючи псевдокод.
3. Оволодіти методами перевірки та оптимізації псевдокоду.
4. Підвищити здатність абстрактного та логічного мислення через процес побудови алгоритмів.

Основне завдання цієї лабораторної роботи - вивчити, як ефективно використовувати псевдокод для планування, опису та візуалізації алгоритмів перед їх реалізацією на конкретній мові програмування.

Псевдокод 1

Функція чи є простим числом:

n-натуральні числа

1. Вхідні данні(числа n)
2. Якщо ≤ 1
 - Не просте
3. Якщо $n=2$,
 - Просте
4. Якщо «2,3»-прості числа, то від 3 з кроком 2;
 - Прості
5. Якщо число ділиться на 2 без залишку
 - не просте Кінець

Перевірка

Початок

1. Вхідні дані(1, 2,5);
2. 1 не просте;
3. 2 просте ;
4. $3+2=5$ просте;

Кінець

Опис: Задача була така щоб визначити чи є число простим. Я спочатку про думав який повинен бути результат і алгоритм. Я написав псевдокод який може визначити декількома способами. І була зроблена перевірка чи вірно він працює. Ось покращений псевдокод для функції перевірки, чи є число простим:

```
Перевірка:
Початок
    Ввести числа: 1, 2, 5
    Для 1:
        Вивести "не просте"
    Для 2:
        Вивести "просте"
    Для 5:
        Виклик функції: чи_є_простим(5)
        Вивести "просте"
Кінець
```

Рисунок №1. Це оптимізований псевдокод

```
Функція чи_є_простим(число n):
    Якщо n <= 1:
        Повернути "не просте"
    Якщо n = 2:
        Повернути "просте"
    Якщо n = 3:
        Повернути "просте"
    Якщо n парне:
        Повернути "не просте"
    Для i від 3 до  $\sqrt{n}$  з кроком 2:
        Якщо n ділиться на i без залишку:
            Повернути "не просте"
    Повернути "просте"
```

Рисунок №2.Перевірка

Опис: Цей псевдокод логічно структурований і забезпечує ефективну перевірку простоти чисел. Він оптимізований тим що коли в першому був пункт «Якщо «2,3»-прості числа, від 3 з кроком 2» , а в цьому коді для більших чисел які будуть знаходитися з під кореня і якщо ділиться залишком то просте.

Псевдокод 4. Який знаходить дублікати

Початок

1. Вхідні дані (Рядок);
2. Створити (Пуста порожнина)для зберігання символів для кожного символу в рядку ;
 - Якщо в рядку є повторювання;

Повернути істина (знайдено повторення);

- Якщо повторювання в наборі не має додаємо його ;

Вивести (Ряд без повторювань);

Кінець

Перевірка 1

Початок

1. Вхідні дані(Привіт);
2. Ряд без повторень

Кінець

Перевірка 2

Початок

1. Вхідні дані(1.1.2.2.3)
2. Істина(1,2) повторення;

Кінець

Опис: Цей алгоритм простий і для цього я зробив дві перевірки на два випадки . Пуста порожнина туди будуть додаватися повторення якщо вони будуть. Якщо повторень не має то код просто виводить що «без повторень».

Псевдокод 2 Найти найбільший дільник

а,б- числа

N-натуральне число

X_n -дільники

Початок

1. Вхідні дані (а, с);
2. Якщо а і с \in N то;
 - Тоді знайти дільники D(а і с) які діляться без залишку;
Вивести результат ($x_1, x_2, x_3, \dots, x_n$);
- Знайти всі додатні дільники для (а і с);
Вивести результат ($x_1, x_2, x_3, \dots, x_n$);
3. Знаходимо найбільше число серед ($x_1, x_2, x_3, \dots, x_n$) для (а і с);
Вивести результат найбільше (x_n);

Кінець

Перевірка

Початок

1. Вхідні дані(3,6);
2. 3 і 6 \in N;
- Результат (1,3,2,3,6,1);
- Результат(1,3,2,3,6,1);
- Результат (3);

Кінець

Опис: Спочатку я хотів зробити свій алгоритм, але знайшов у інтернет ресурсі Евклідов спосіб. І я описав його таким чином. Я так думаю що це на швидший спосіб.

Псевдокод. Обчислення факторіалу

a-це будь-яке натуральне число

x-це числа які треба помножити

n-число яке вийде

Початок

Введіть Число(a);

1. Якщо $a=0$;
 - Вивести: $0!=1$;
2. Якщо $a>0$;
3. Для x від 1 до a, множаться;
 - вивести n;

Кінець

Перевірка;

Початок

1. Введіть Число(4);
 2. $4! \neq 0$;
 3. Для 1 2 3 4;
- $1*2*3*4=24$:

Кінець

Опис: Цей код рахує факторіал числа тут алгоритм не важкий. І доволі простий. Він просто бере множе всі числа від 1 до заданого і виходить результат. Перевіркою доведено.

Висновок

Дана лабораторна робота мене навчила як писати прості алгоритми. Вникали труднощі з написання і-за недостатнього досвіду в прописуванні алгоритмів. Навчився оптимізувати код.

Також розвинув свої навички в логічному мисленні. Але над створенням таких типових завдань треба більше досвіду.