

Лабораторна робота за замовчуванням (30 б)

Самостійно створити програмний інструмент, що надає можливість користувачу:

Пункт 1. Задавати поведінку багатопоточної (консольної) програми з $1 \leq N \leq 100$ потоками за допомогою набору з N блок-схем, кожна з яких відповідає одному потоку. Кожна блок-схема може містити до 100 блоків. Користувач повинен мати можливість створювати, редагувати та зберігати у файлі такі блок-схеми за допомогою графічного інтерфейсу; кожна блок-схема задає послідовний алгоритм, що виконує відповідний потік. Орієнтовані графи блок-схем можуть містити цикли (маються на увазі цикли у графі, а не цикли while).

Дії, що допускаються у блок-схемах (що задають поведінки окремих потоків) мають включати:

* присвоєння вигляду $V1 = V2$, що розуміється як “значення $V2$ записати у $V1$ ”, де $V1, V2$ – спільні змінні 32-бітного цілочисельного типу (змінні у спільній пам’яті, що є доступними усім потокам; користувач повинен мати можливість використовувати до 100 таких змінних в одній блок-схемі)

* присвоєння вигляду $V = C$, що розуміється як “значення C записати у V ”, де V – спільна змінна, C – (десятковий) літерал, що задає ціле число (константу) в діапазоні $0 \dots 2^{31} - 1$

* команди вигляду $\text{INPUT } V$, що розуміються як “прочитати цілочисельне значення в діапазоні $0 \dots 2^{31} - 1$ зі стандартного вводу та записати його у V ”, де V – спільна змінна

* команди вигляду $\text{PRINT } V$, що розуміються як “вивести значення V на стандартний вивід”, де V – спільна змінна.

Умови, допустимі у блоках розгалуження мають включати:

* порівняння вигляду $V == C$, що розуміється як “значення V рівне значенню C ”, де V – спільна змінна, C – літерал

* порівняння вигляду $V < C$, що розуміється як “значення V менше значення C ”, де V – спільна змінна, C – літерал.

Пункт 2. Виконувати автоматичну трансляцію набору $1 \leq N \leq 100$ блок-схем (див. пункт 1) у вихідний код багатопоточної програми з N потоками на одній з мов програмування C, C++, C#, Java, Python (на Ваш вибір).

Пункт 3. Виконувати автоматичне тестування набору блок-схем (заданого згідно пункту 1) на наборі тестів (*test set*), заданому користувачем. Кожний тест (*test case*) є парою з вхідних даних, що подаються на стандартний ввід та вихідних даних, що очікуються на стандартному виводі після завершення роботи усіх потоків. При тестуванні вважати, що семантика набору блок-схем відповідає семантиці програми, що є результатом її трансляції згідно пункту 2. Якщо відповідно до цієї семантики набір блок-схем задає недетерміновану програму (вихідні дані якої не визначаються однозначно вхідними даними, що можливо, наприклад, якщо два потоки незалежно виводять 0 та 1), то надати користувачу можливість багаторазового тестування програми на одному й тому ж тесті з перебором **різних** виконань (без повторів), що відповідають одним і тим же вхідним даним. Надати користувачу можливість достроково перервати процес виконання цього перебору, ввести ціле число $1 \leq K \leq 20$, та дізнатися відсоток усіх варіантів виконання програми, що складаються з не більше як K операцій та відповідають вхідним даним поточного тесту, для яких коректність виводу програми було перевірено до даного моменту в процесі тестування.