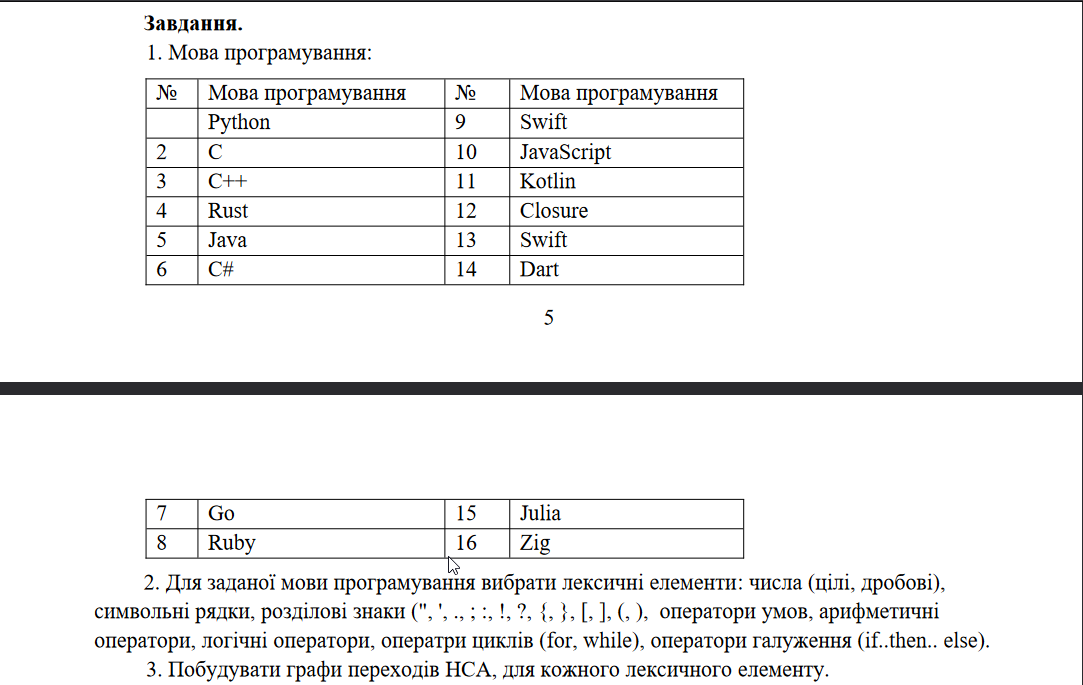
Завдання



#include <iostream>

#include <unordered\_map>

#include <unordered\_set>

#include <vector>

#include <string>

using namespace std;

// Структура для представлення стану в графі переходів НСА

struct State {

    unordered\_map<char, unordered\_set<int>> transitions; // переходи за символом

    // Додати перехід до іншого стану

    void addTransition(char input, int nextState) {

        transitions[input].insert(nextState);

    }

    // Отримати стани, в які можна перейти за певним символом

    unordered\_set<int>& getTransitions(char input) {

        return transitions[input];

    }

};

// Клас, що представляє НСА для певного лексичного елемента

class NFA {

private:

    vector<State> states;

    int startState;

    unordered\_set<int> finalStates;

public:

    NFA() {

        // Створення початкового стану

        states.emplace\_back();

        startState = 0;

        // Створення стану, який є фінальним

        states.emplace\_back();

        finalStates.insert(1);

        // Додавання переходів за символами

        states[0].addTransition('i', 1);

        states[1].addTransition('f', 1);

    }

    // Перевірка, чи може НСА прийняти дану послідовність символів

    bool accepts(const string& input) {

        unordered\_set<int> currentStates = { startState }; // початковий стан

        for (char ch : input) {

            unordered\_set<int> nextStates;

            for (int state : currentStates) {

                // Додати всі наступні стани для кожного символу вводу

                auto& transitions = states[state].getTransitions(ch);

                nextStates.insert(transitions.begin(), transitions.end());

            }

            currentStates = move(nextStates);

        }

        // Перевірити, чи є хоча б один фінальний стан

        for (int state : currentStates) {

            if (finalStates.count(state)) {

                return true;

            }

        }

        return false;

    }

};

int main() {

    NFA nfa;

    // Перевірка послідовностей на відповідність лексичному елементу "if"

    cout << "Перевірка 'if': " << (nfa.accepts("if") ? "Прийнято" : "Не прийнято") << endl;

    cout << "Перевірка 'else': " << (nfa.accepts("else") ? "Прийнято" : "Не прийнято") << endl;

    cout << "Перевірка 'ifelse': " << (nfa.accepts("ifelse") ? "Прийнято" : "Не прийнято") << endl;

    return 0;

}

