COMPILER DESIGN LAB 2

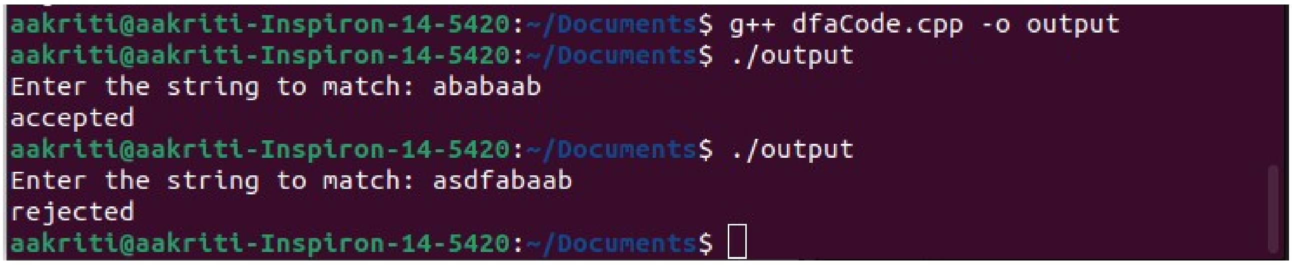
BY~
AAKRITI MEHROTRA
22BCE1954

REGULAR EXPRESSION CONSIDERED: (a|b)*aab

CODE :

```
#include <iostream>
#include <unordered_map>
#include <set>
#include <string>

class DFA {
public:
    std::unordered_map<int, std::unordered_map<char, int>> transitions;
    std::set<int> acceptingStates;
    int startState;

    DFA() : startState(0) {}

    void addTransition(int fromState, char symbol, int toState) {
        transitions[fromState][symbol] = toState;
    }

    void addAcceptingState(int state) {
        acceptingStates.insert(state);
    }

    bool isAccepting(int state) const {
        return acceptingStates.find(state) != acceptingStates.end();
    }

    bool processString(const std::string& str) const {
        int currentState = startState;
        for (char symbol : str) {
            if (transitions.at(currentState).find(symbol) != transitions.at(currentState).end()) {
                currentState = transitions.at(currentState).at(symbol);
            } else {
                return false;
            }
        }
        return isAccepting(currentState);
    }
};

DFA constructDFA() {
    DFA dfa;
    // State 0 transitions (initial state)
    dfa.addTransition(0, 'a', 1);
    dfa.addTransition(0, 'b', 0);

    // State 1 transitions (we've seen 'a')
    dfa.addTransition(1, 'a', 2);
    dfa.addTransition(1, 'b', 0);

    // State 2 transitions (we've seen 'aa')
    dfa.addTransition(2, 'a', 2);
    dfa.addTransition(2, 'b', 3);

    // State 3 transitions (we've seen 'aab')
    dfa.addTransition(3, 'a', 1);
    dfa.addTransition(3, 'b', 0);

    // State 3 is the accepting state
    dfa.addAcceptingState(3);

    return dfa;
}

int main() {
    DFA dfa = constructDFA();
    std::string inputString;

    std::cout << "Enter the string to match: ";
    std::cin >> inputString;

    if (dfa.processString(inputString)) {
        std::cout << "accepted" << std::endl;
    } else {
        std::cout << "rejected" << std::endl;
    }

    return 0;
}
```

OUTPUT



```
aakriti@aakriti-Inspiron-14-5420:~/Documents$ g++ dfaCode.cpp -o output
aakriti@aakriti-Inspiron-14-5420:~/Documents$ ./output
Enter the string to match: abababb
accepted
aakriti@aakriti-Inspiron-14-5420:~/Documents$ ./output
Enter the string to match: asdfabaab
rejected
aakriti@aakriti-Inspiron-14-5420:~/Documents$ 
```