**Assignment 3**

**Problem Statement:**  
Perform parsing of a **Family Tree** using a **Knowledge Base**.

**Theory**

**1. Knowledge-Based Systems**

* A **Knowledge-Base (KB)** is a collection of facts and rules about a domain.
* In AI, KBs are often used for reasoning and inference.
* Example: In a family tree, facts like *“John is the parent of Mary”* or *“Mary is a sibling of Mark”* can be stored.

**2. Family Tree Representation**

* A **family tree** is a hierarchical structure that represents relationships among individuals.
* Key relationships:
  + **Parent-Child**
  + **Siblings**
* These relationships can be represented using a **map (dictionary)**, where each parent maps to a list of their children.

**3. Parsing in Knowledge Base**

* **Parsing** means extracting and processing relationships from stored knowledge.
* For the family tree:
  + Parse data to find children of a parent.
  + Check if one person is parent of another.
  + Determine if two people are siblings.

**Algorithm Steps**

1. **Add Relations:** Insert parent-child facts into the knowledge base (map).
2. **Show Children:** Retrieve all children of a given parent.
3. **Check Parent:** Verify if a person is parent of another.
4. **Check Sibling:** Look for a common parent between two individuals.
5. Perform queries on the knowledge base to display family relationships.

**Code (C++ Implementation)**

#include <iostream>

#include <map>

#include <vector>

#include <algorithm>

using namespace std;

map<string, vector<string>> family;

// Function to add parent-child relation

void addRelation(string parent, string child) {

family[parent].push\_back(child);

}

// Function to display children of a parent

void showChildren(string parent) {

if (family.find(parent) == family.end()) {

cout << parent << " has no children.\n";

return;

}

cout << "Children of " << parent << ": ";

for (auto &c : family[parent]) cout << c << " ";

cout << endl;

}

// Function to check if A is parent of B

void isParent(string A, string B) {

if (family.find(A) != family.end() &&

find(family[A].begin(), family[A].end(), B) != family[A].end()) {

cout << A << " is parent of " << B << endl;

} else {

cout << A << " is NOT parent of " << B << endl;

}

}

// Function to check if A and B are siblings

void isSibling(string A, string B) {

for (auto &p : family) {

auto &kids = p.second;

if (find(kids.begin(), kids.end(), A) != kids.end() &&

find(kids.begin(), kids.end(), B) != kids.end() &&

A != B) {

cout << A << " and " << B << " are siblings\n";

return;

}

}

cout << A << " and " << B << " are NOT siblings\n";

}

int main() {

// Adding family relations

addRelation("John", "Mary");

addRelation("John", "Mark");

addRelation("Alice", "John");

addRelation("Alice", "Linda");

addRelation("Linda", "Sam");

// Display children

showChildren("John");

showChildren("Alice");

// Check parent relations

isParent("John", "Mary");

isParent("Alice", "Sam");

// Check siblings

isSibling("Mary", "Mark");

isSibling("John", "Linda");

return 0;

}

**Sample Output**

Children of John: Mary Mark

Children of Alice: John Linda

John is parent of Mary

Alice is NOT parent of Sam

Mary and Mark are siblings

John and Linda are NOT siblings

**Conclusion**

* The **family tree parsing system** was successfully implemented using a **knowledge-base**.
* Relationships such as **parent-child** and **siblings** were derived using simple reasoning functions.
* This assignment demonstrates how **knowledge representation** and **parsing** can be applied in AI for real-world relationship modeling.