123103054

Nishant

IT – ‘A4’

Washing mashine problem

code -

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

float triangle(float x, float a, float b, float c) {
  if (x <= a || x >= c) return 0;
  else if (x == b) return 1;
  else if (x > a && x < b) return (x - a) / (b - a);
  else return (c - x) / (c - b);
}

struct FuzzyValue {
  float low, medium, high;
};

FuzzyValue fuzzifyDirtiness(float dirtiness) {
  FuzzyValue val;
  val.low = triangle(dirtiness, 0, 0, 50);
  val.medium = triangle(dirtiness, 25, 50, 75);
  val.high = triangle(dirtiness, 50, 100, 100);
  return val;
}

FuzzyValue fuzzifyFabric(float fabric) {
  FuzzyValue val;
  val.low = triangle(fabric, 0, 0, 50);
  val.medium = triangle(fabric, 25, 50, 75);
  val.high = triangle(fabric, 50, 100, 100);
  return val;
}

struct Output {
  float washTime;
  float waterLevel;
};

vector<pair<float, Output>> applyRules(FuzzyValue dirt, FuzzyValue fabric) {
  vector<pair<float, Output>> rules;

  // Rule 1: If dirt is high AND fabric is heavy THEN time = long, water = high
  rules.push_back({min(dirt.high, fabric.high), {50, 90}});

  // Rule 2: If dirt is high AND fabric is medium THEN time = long, water = medium
  rules.push_back({min(dirt.high, fabric.medium), {45, 70}});

  // Rule 3: If dirt is high AND fabric is low THEN time = medium, water = medium
  rules.push_back({min(dirt.high, fabric.low), {40, 60}});
```

```
    // Rule 4: If dirt is medium AND fabric is high THEN time = medium, water = high
    rules.push_back({min(dirt.medium, fabric.high), {40, 85}});

    // Rule 5: If dirt is medium AND fabric is medium THEN time = medium, water =
medium
    rules.push_back({min(dirt.medium, fabric.medium), {35, 65}});

    // Rule 6: If dirt is medium AND fabric is low THEN time = short, water = medium
    rules.push_back({min(dirt.medium, fabric.low), {30, 60}});

    // Rule 7: If dirt is low AND fabric is high THEN time = short, water = medium
    rules.push_back({min(dirt.low, fabric.high), {25, 60}});

    // Rule 8: If dirt is low AND fabric is medium THEN time = short, water = low
    rules.push_back({min(dirt.low, fabric.medium), {20, 40}});

    // Rule 9: If dirt is low AND fabric is low THEN time = very short, water = low
    rules.push_back({min(dirt.low, fabric.low), {15, 30}});

    return rules;
}

Output defuzzify(const vector<pair<float, Output>>& rules) {
    float sumWeight = 0;
    float washSum = 0;
    float waterSum = 0;

    for (auto& rule : rules) {
        float weight = rule.first;
        washSum += weight * rule.second.washTime;
        waterSum += weight * rule.second.waterLevel;
        sumWeight += weight;
    }

    Output result;
    if (sumWeight == 0) {
        result.washTime = 0;
        result.waterLevel = 0;
    } else {
        result.washTime = washSum / sumWeight;
        result.waterLevel = waterSum / sumWeight;
    }

    return result;
}

int main() {
    float dirtiness, fabric;

    dirtiness = 70;
    fabric = 10;
```

```
    FuzzyValue dirtFuzz = fuzzifyDirtiness(dirtiness);
    FuzzyValue fabricFuzz = fuzzifyFabric(fabric);

    vector<pair<float, Output>> rules = applyRules(dirtFuzz, fabricFuzz);
    Output result = defuzzify(rules);

    cout << "Input:" << endl;
    cout << "Degree of Dirtiness: " << dirtiness << endl;
    cout << "Fabric Type: " << fabric << endl;

    cout << "\nOutput:" << endl;
    cout << "Wash Time: " << result.washTime << " minutes" << endl;
    cout << "Water Level: " << result.waterLevel << "%" << endl;

    return 0;
}
```

OUTPUT -

```
Input:
Degree of Dirtiness: 70
Fabric Type: 10

Output:
Wash Time: 36.6667 minutes
Water Level: 60%
```