Negative Wt Cycle Detection in C++

```
#include <iostream>
#include <vector>
#include <climits>

using namespace std;

struct Edge {
   int u, v, weight;
};

bool isNegativeWeightCycle(int n, vector<Edge>& edges)
{
   vector<int> dist(n, INT_MAX);
   dist[0] = 0; // Starting from vertex 0

   // Relaxation process
   for (int i = 0; i < n - 1; ++i) {
      for (const auto& edge : edges) {
         if (dist[edge.u] != INT_MAX && dist[edge.u] +
edge.weight < dist[edge.v]) {
            dist[edge.v] = dist[edge.u] + edge.weight;
         }
      }
   }

   // Checking for negative weight cycles
   for (const auto& edge : edges) {
      if (dist[edge.u] != INT_MAX && dist[edge.u] +
edge.weight < dist[edge.v]) {
         return true; // Negative weight cycle detected
      }
   }

   return false; // No negative weight cycle found
}

int main() {
   // Hardcoded input
   int n = 3; // Number of vertices
   int m = 3; // Number of edges
   vector<Edge> edges = {{0, 1, -1}, {1, 2, -4}, {2, 0, 3}}; //
Edges with (u, v, weight)

   if (isNegativeWeightCycle(n, edges)) {
      cout << "1\n"; // Negative weight cycle detected
   } else {
      cout << "0\n"; // No negative weight cycle found
   }

   return 0;
}
```

Step-by-Step Execution

Input Edges:

Edge 1: (0 → 1, weight = -1)
Edge 2: (1 → 2, weight = -4)
Edge 3: (2 → 0, weight = 3)

Initial State:

dist = [0, INT_MAX, INT_MAX]

Relaxation Process (n-1 = 2 times):

- **Iteration 1:**
  - Relax Edge (0 → 1, weight = -1):

    dist[1] = min(INT_MAX, dist[0] + (-1)) = min(INT_MAX, 0 + (-1)) = -1
    dist = [0, -1, INT_MAX]
  - Relax Edge (1 → 2, weight = -4):

    dist[2] = min(INT_MAX, dist[1] + (-4)) = min(INT_MAX, -1 + (-4)) = -5
    dist = [0, -1, -5]
  - Relax Edge (2 → 0, weight = 3):

    dist[0] = min(0, dist[2] + 3) = min(0, -5 + 3) = -2
    dist = [-2, -1, -5]
- **Iteration 2:**
  - Relax Edge (0 → 1, weight = -1):

    dist[1] = min(-1, dist[0] + (-1)) = min(-1, -2 + (-1)) = -3
    dist = [-2, -3, -5]
  - Relax Edge (1 → 2, weight = -4):

    dist[2] = min(-5, dist[1] + (-4)) = min(-5, -3 + (-4)) = -7
    dist = [-2, -3, -7]
  - Relax Edge (2 → 0, weight = 3):

    dist[0] = min(-2, dist[2] + 3) = min(-2, -7 + 3) = -4
    dist = [-4, -3, -7]

Check for Negative Weight Cycles:

- Try relaxing all edges once more:

<table>
<tr><td></td><td>○ Relax Edge (0 → 1, weight = -1):
dist[1] = min(-3, dist[0] + (-1)) = min(-3, -4 + (-1)) = -5
○ At this point, the distance to vertex 1 changes, which means a negative weight cycle exists.
<b>Output:</b>
• Since a negative weight cycle is detected:
1</td></tr>
<tr><td colspan="2">Output:-
1</td></tr>
</table>