```cpp
#include <iostream>
#include <vector>
#include <map>
#include <functional>
#include <algorithm>


std::vector<int> A_star(const std::map<int, std::vector<int>> &graph, const
std::map<std::pair<int, int>, int> &wages,
                        const std::map<int, std::pair<int, int>>
&coordinates, int start, int finish) {

    std::vector<int> To = {}; // wierzcholki zamkniete
    std::vector<int> Tc = {}; // wierzcholki otwarte
    std::map<int, int> actual_costs = {}; // g[u] - aktualny koszt
    std::map<int, int> estimated_full_costs = {}; // h[u] - estymacja kosztu
    std::map<int, int> estimated_part_costs = {}; // f[u] = g[u] + h[u] -
estymacja osigniecia kosztu z s do k przez u
    std::map<int, int> previous_node = {}; // poprzednik wierzcholka u
    std::vector<int> result = {}; // najktorsza sciezka


    return {};
}


int main() {
    // graf
    std::map<int, std::vector<int>> g1 = {
            {0, {1, 2}},
            {1, {4, 5}},
            {2, {3, 4}},
            {3, {6, 7}},
            {4, {3, 5, 6}},
            {5, {4, 6}},
            {6, {7, 8}},
            {7, {8, 9}},
            {8, {7, 9}},
            {9, {}}
    };

    // funckcja wag
    std::map<std::pair<int, int>, int> w1 = {
            {{0, 1}, 5},
            {{0, 2}, 6},
            {{1, 4}, 4},
            {{1, 5}, 5},
            {{2, 3}, 5},
            {{2, 4}, 3},
            {{3, 6}, 9},
            {{3, 7}, 9},
            {{4, 3}, 3},
            {{4, 5}, 5},
            {{4, 6}, 9},
            {{5, 4}, 5},
            {{5, 6}, 8},
            {{6, 7}, 4},
```

```
            {{6, 8}, 5},
            {{7, 8}, 4},
            {{7, 9}, 8},
            {{8, 7}, 4},
            {{8, 9}, 6}
    };

    // wspolrzedne
    std::map<int, std::pair<int, int>> c1 = {
            {0, {1,  1}},
            {1, {4,  2}},
            {2, {2,  5}},
            {3, {5,  8}},
            {4, {5,  5}},
            {5, {8,  1}},
            {6, {13, 7}},
            {7, {13, 11}},
            {8, {17, 10}},
            {9, {18, 16}}
    };

    auto result = A_star(g1, w1, c1, 0, 9);

    return 0;
}
```