include <iostream>

include <vector>

include <string>

double cal(const std::string& fun, const std::vector<double>& params) {

if (params.empty()) {

throw std::invalid\_argument("Parameter list is empty");

}

double result = params[0];

if (fun == "add") {

for (size\_t i = 1; i < params.size(); ++i) {

result += params[i];

}

}

else if (fun == "minus") {

for (size\_t i = 1; i < params.size(); ++i) {

result -= params[i];

}

}

else {

throw std::invalid\_argument("Invalid operation");

}

return result;

}

int main() {

try {

std::vector<double> numbers = {10, 20, 30, 40};

double sum = cal("add", numbers);

std::cout << "Sum: " << sum << std::endl;

double difference = cal("minus", numbers);

std::cout << "Difference: " << difference << std::endl;

}

catch (const std::invalid\_argument& e) {

std::cerr << "Error: " << e.what() << std::endl;

return 1;

}

return 0;

}