#include <iostream>

#include <vector>

#include <algorithm>

#include <map>

double calculateMean(const std::vector<int>& numbers) {

int sum = 0;

for (int num : numbers) {

sum += num;

}

return static\_cast<double>(sum) / numbers.size();

}

double calculateMedian(const std::vector<int>& numbers) {

std::vector<int> sortedNumbers = numbers;

std::sort(sortedNumbers.begin(), sortedNumbers.end());

int size = numbers.size();

if (size % 2 == 0) {

return (sortedNumbers[size / 2 - 1] + sortedNumbers[size / 2]) / 2.0;

} else {

return sortedNumbers[size / 2];

}

}

std::vector<int> calculateMode(const std::vector<int>& numbers) {

std::map<int, int> frequencyMap;

for (int num : numbers) {

frequencyMap[num]++;

}

std::vector<int> modes;

int maxFrequency = 0;

for (const auto& entry : frequencyMap) {

if (entry.second > maxFrequency) {

maxFrequency = entry.second;

modes.clear();

modes.push\_back(entry.first);

} else if (entry.second == maxFrequency) {

modes.push\_back(entry.first);

}

}

return modes;

}

int main() {

std::vector<int> numbers;

int num;

std::cout << "Enter the numbers (enter -1 to finish):\n";

while (true) {

std::cin >> num;

if (num == -1) {

break;

}

numbers.push\_back(num);

}

double mean = calculateMean(numbers);

double median = calculateMedian(numbers);

std::vector<int> modes = calculateMode(numbers);

std::cout << "Mean: " << mean << std::endl;

std::cout << "Median: " << median << std::endl;

std::cout << "Mode(s): ";

for (int mode : modes) {

std::cout << mode << " ";

}

std::cout << std::endl;

return 0;

}