Day-48 of the #101 days of the coding challenge----------

Problem:-

Given an array of integers, calculate the ratios of its elements that are *positive*, *negative*, and *zero*.

Print the decimal value of each fraction on a new line with places after the decimal.

Note: This challenge introduces precision problems. The test cases are scaled to six decimal

places, though answers with absolute error of up to are acceptable.

There are elements, two positive, two negative and one zero. Their ratios are , and . Results are

printed as:

TDIN Function

----- --------

6 arr[] size n = 6

-4 3 -9 0 4 1 arr = [-4, 3, -9, 0, 4, 1]

Sample Output

0.500000

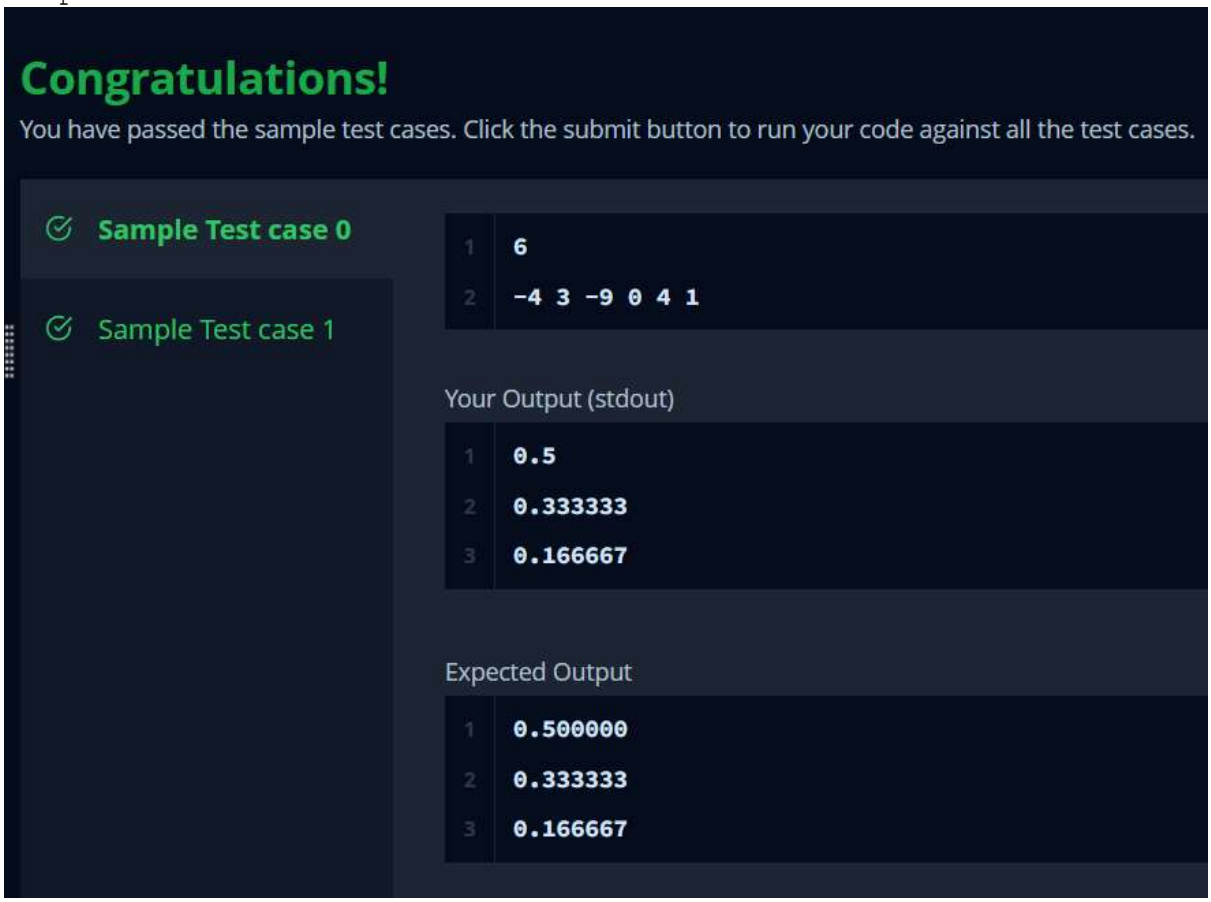
0.333333

0.166667

Code:-

```
void plusMinus(vector<int> arr) {

    int i, posCount = 0, negCount = 0, zeroCount = 0;
    int length = arr.size(); // length of the array

    for(i = 0; i<length; i++){

        if(arr[i]>0)
           posCount++;
        if(arr[i]<0)
           negCount++;
        if(arr[i] == 0)
           zeroCount++;
    }

    double positiveRatio = static_cast<double>(posCount)/length;
// type casting into double
    double negativeRation = static_cast<double>(negCount)/length;
    double zeroRatio = static_cast<double>(zeroCount)/length;

    cout<<positiveRatio<<"\n"<<negativeRation<<"\n"<<zeroRatio }
```

Output:-



Problem:-

Given five positive integers, find the minimum and maximum values that can be calculated by summing exactly four of the five integers. Then print the respective minimum and maximum values as a single line of two space-separated long integers

Sample Input

1 2 3 4 5

Sample Output

10 14

Code:-

```
void miniMaxSum(vector<int> arr) {

    long long int minSum = LLONG_MAX; // Initialize minSum to the
 maximum possible value
    long long int maxSum = LLONG_MIN; // Initialize maxSum to the
 minimum possible value
    long long int totalSum = 0;

    int length = arr.size();
```

```
    for(int i = 0; i < length; i++) {
        totalSum += arr[i];
        minSum = min(minSum, static_cast<long long int>(arr[i]));
        maxSum = max(maxSum, static_cast<long long int>(arr[i]));
    }

    long long int minSumResult = totalSum - maxSum;
    long long int maxSumResult = totalSum - minSum;

    cout << minSumResult << " " << maxSumResult;
}
```

Output:-