#include <assert.h>

#include <limits.h>

#include <math.h>

#include <stdbool.h>

#include <stddef.h>

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char\* readline();

char\*\* split\_string(char\*);

// Complete the plusMinus function below.

void plusMinus(int arr\_count, int\* arr) {

float new\_arr[arr\_count];

int i;

float temp;

float count;

count = arr\_count;

float pos,neg,ze;

pos = 0.0f;

neg = 0.0f;

ze = 0.0f;

for(i=0;i<arr\_count;i++)

{

temp = arr[i];

new\_arr[i] = temp/count;

}

for(i=0;i<arr\_count;i++)

{

//printf("%f",new\_arr[i]);

if(new\_arr[i]<0)

{

neg = neg + 1/count;

}

if(new\_arr[i]==0)

{

ze = ze + 1/count;

}

if(new\_arr[i]>0)

{

pos = pos + 1/count;

}

}

printf("%f",pos);

printf("\n%f",neg);

printf("\n%f",ze);

}

int main()

{

char\* n\_endptr;

char\* n\_str = readline();

int n = strtol(n\_str, &n\_endptr, 10);

if (n\_endptr == n\_str || \*n\_endptr != '\0') { exit(EXIT\_FAILURE); }

char\*\* arr\_temp = split\_string(readline());

int\* arr = malloc(n \* sizeof(int));

for (int i = 0; i < n; i++) {

char\* arr\_item\_endptr;

char\* arr\_item\_str = \*(arr\_temp + i);

int arr\_item = strtol(arr\_item\_str, &arr\_item\_endptr, 10);

if (arr\_item\_endptr == arr\_item\_str || \*arr\_item\_endptr != '\0') { exit(EXIT\_FAILURE); }

\*(arr + i) = arr\_item;

}

int arr\_count = n;

plusMinus(arr\_count, arr);

return 0;

}

char\* readline() {

size\_t alloc\_length = 1024;

size\_t data\_length = 0;

char\* data = malloc(alloc\_length);

while (true) {

char\* cursor = data + data\_length;

char\* line = fgets(cursor, alloc\_length - data\_length, stdin);

if (!line) { break; }

data\_length += strlen(cursor);

if (data\_length < alloc\_length - 1 || data[data\_length - 1] == '\n') { break; }

size\_t new\_length = alloc\_length << 1;

data = realloc(data, new\_length);

if (!data) { break; }

alloc\_length = new\_length;

}

if (data[data\_length - 1] == '\n') {

data[data\_length - 1] = '\0';

}

data = realloc(data, data\_length);

return data;

}

char\*\* split\_string(char\* str) {

char\*\* splits = NULL;

char\* token = strtok(str, " ");

int spaces = 0;

while (token) {

splits = realloc(splits, sizeof(char\*) \* ++spaces);

if (!splits) {

return splits;

}

splits[spaces - 1] = token;

token = strtok(NULL, " ");

}

return splits;

}