C++ Fundamentals: Exam

The following tasks should be submitted to the SoftUni Judge system, which will be open starting Sunday, 14 January 2018, 09:00 (in the morning) and will close the same day at 15:00. Submit your solutions here: https://judge.softuni.bg/Contests/Compete/Index/907.

For this exam, the code for each task should be a single C++ file, the contents of which you copy-paste into the Judge system.

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **`int`** and any floating-point input can be stored in **`double`**. On the Judge system, a C++ **`int`** is a **`32-bit`** signed integer and a C++ **`double`** is a **`64-bit`** IEEE754 floating point number.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 1 – Extremes (Exam-Task-1-Extremes)

Our hero Ben Dover (*When did he become our hero? How should I know, I just talk about C++ here…*) has been given a statistics task by his employer. He has to calculate the average income of the village he lives in. But he can't just compute the average of all the incomes, because some people in the village have very high incomes and others have very low incomes (some even have credit, making their incomes effectively less than 0) and are not considered part of the average. Such values, which are not included in a statistics calculation, are often called "outliers" in statistics. So, Ben Dover needs to calculate the average of only those incomes, which are a certain "distance" away from the minimum and the maximum income in the village, i.e. calculate the average of the incomes that are not outliers.

You are given a sequence of integer numbers (the incomes of the people in the village) and a non-negative value called an "extremes distance".

Any value within the "extremes distance" from one of the two extremes (the minimum or the maximum) is considered part of the "outliers" (this means that the minimum and maximum are also outliers and always will be, regardless of the "extremes distance").

Your task is to calculate the average of the numbers which are not outliers, i.e. the average of the numbers which are not within the extremes distance of neither the minimum nor the maximum.

Input

The first line of the standard input will contain a single positive integer number **D** – the "extremes distance"

The second line of the standard input will contain a single positive integer number **N** – the number of incomes.

The third line of the standard input will contain exactly **N** integer values, separated by single spaces – the values (of the incomes).

Output

A single line, containing a single number – the average income, not including the outliers. Note that this value is not necessarily an integer.

Restrictions

`0 <= D <= 100`

`0 < N <= 500`;

There will always be a least 1 value in the input which is NOT an outlier (the minimum and the maximum value will always be different numbers and their absolute difference will be larger than **D**)

The input will be such that the output will have no more than 3 digits after the decimal point and answers will be between **`-10000`** and **`10000`** (i.e. simply printing the result without special formatting should be ok).

The total running time of your program should be no more than **`0.1s`**

The total memory allowed for use by your program is **16MB**

Example I/O

Example Input	Expected Output	Explanation

1
6
4 7 2 1 3 7
	3.5	Min is 1, max is 7, and within a distance of 1 we have 2 and 7, 7 respectively. That means that 1, 2, 7, 7 (note there are two incomes with a value of 7) are outliers, leaving us with 3 and 4 for the calculation – their average is 3.5

6
10
-40 -35 -30 -20 5 10 15 15 15 20
	-8.75	Min is -40, max is 20, within a distance of 6 we have -40, -35 for min and 15, 15, 15, 20 for max. We are left with -30, -20-, 5, 10.
		The average of -30, -20, 5, 10 is -35/4 = -8.75