Programs

Birthday Cake Candles

You are in-charge of the cake for your niece's birthday and have decided the cake will have one candle for each year of her total age. When she blows out the candles, she'll only be able to blow out the tallest ones.

For example, if your niece is turning 4 years old, and the cake will have 4 candles of height 3, 2, 1, 3, she will be able to blow out 2 candles successfully, since the tallest candle is of height 3 and there are 2 such candles.

Complete the function Given the height of each individual candle, find and print the number of candles she can successfully blow out.

Input Format

The first line contains a single integer, n, denoting the number of candles on the cake.
The second line contains n space-separated integers, where each integer i describes the height of candle i.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq height_i \leq 10^7$

Output Format

Print the number of candles the can be blown out on a new line.

Sample Input 0

```
4
3 2 1 3
```

Sample Output 0

```
2
```

Explanation 0

We have one candle of height 1, one candle of height 2, and two candles of height 3. Your niece only blows out the tallest candles, meaning the candles where $height = 3$. Because there are 2 such candles, we print 2 on a new line.

Grading Students

HackerLand University has the following grading policy:

- Every student receives a $grade$ in the inclusive range from 0 to 100.
- Any $grade$ less than 40 is a failing grade.

Sam is a professor at the university and likes to round each student's $grade$ according to these rules:

- If the difference between the $grade$ and the next multiple of 5 is less than 3, round $grade$ up to the next multiple of 5.
- If the value of $grade$ is less than 38, no rounding occurs as the result will still be a failing grade.

For example, $grade = 84$ will be rounded to 85 but $grade = 29$ will not be rounded because the rounding would result in a number that is less than 40.

Given the initial value of $grade$ for each of Sam's n students, write code to automate the rounding process. For each $grade_i$, round it according to the rules above and print the result on a new line.

Input Format

The first line contains a single integer denoting n (the number of students).
Each line i of the n subsequent lines contains a single integer, $grade_i$, denoting student i's grade.

Constraints

- $1 \leq n \leq 60$
- $0 \leq grade_i \leq 100$

Output Format

For each $grade_i$ of the n grades, print the rounded grade on a new line.

Sample Input 0

```
4
73
67
38
33
```

Sample Output 0

```
75
67
40
33
```

Explanation 0

ID	Original Grade	Final Grade
1	73	75
2	67	67
3	38	40
4	33	33

1. Student 1 received a 73, and the next multiple of 5 from 73 is 75. Since $75 - 73 < 3$, the student's grade is rounded to 75.

2. Student 2 received a 67, and the next multiple of 5 from 67 is 70. Since $70 - 67 = 3$, the grade will not be modified and the student's final grade is 67.

3. Student 3 received a 38, and the next multiple of 5 from 38 is 40. Since $40 - 38 < 3$, the student's grade will be rounded to 40.

4. Student 4 received a grade below 38, so the grade will not be modified and the student's final grade is 33.

Compare the Triplets

Alice and Bob each created one problem for HackerRank. A reviewer rates the two challenges, awarding points on a scale from 1 to 100 for three categories: *problem clarity*, *originality*, and *difficulty*.

We define the rating for Alice's challenge to be the triplet $A = (a[0], a[1], a[2])$, and the rating for Bob's challenge to be the triplet $B = (b[0], b[1], b[2])$.

Your task is to find their *comparison points* by comparing $a[0]$ with $b[0]$, $a[1]$ with $b[1]$, and $a[2]$ with $b[2]$.

- If $a[i] > b[i]$, then Alice is awarded 1 point.
- If $a[i] < b[i]$, then Bob is awarded 1 point.
- If $a[i] = b[i]$, then neither person receives a point.

Comparison points is the total points a person earned.

Given a and b, determine their respective comparison points.

For example, $a = [1, 2, 3]$ and $b = [3, 2, 1]$. For elements 0, Bob is awarded a point because $a[0] < b[0]$. For the equal elements $a[1]$ and $b[1]$, no points are earned. Finally, for elements 2, $a[2] > b[2]$ so Alice receives a point. Your return array would be $[1, 1]$ with Alice's score first and Bob's second.

Function Description

Complete the function *compareTriplets* in the editor below. It must return an array of two integers, the first being Alice's score and the second being Bob's.

compareTriplets has the following parameter(s):

- *a*: an array of integers representing Alice's challenge rating
- *b*: an array of integers representing Bob's challenge rating

Input Format

The first line contains 3 space-separated integers, $a[0]$, $a[1]$, and $a[2]$, describing the respective values in triplet A.
The second line contains 3 space-separated integers, $b[0]$, $b[1]$, and $b[2]$, describing the respective values in triplet B.

Constraints

- $1 \leq a[i] \leq 100$
- $1 \leq b[i] \leq 100$

Output Format

Return an array of two integers denoting the respective comparison points earned by Alice and Bob.

Sample Input 0

```
5 6 7
3 6 10
```

Sample Output 0

```
1 1
```

Explanation 0

In this example:

- $a = (a[0], a[1], a[2]) = (5, 6, 7)$
- $b = (b[0], b[1], b[2]) = (3, 6, 10)$

Now, let's compare each individual score:

- $a[0] > b[0]$, so Alice receives 1 point.
- $a[1] = b[1]$, so nobody receives a point.
- $a[2] < b[2]$, so Bob receives 1 point.

Alice's comparison score is 1, and Bob's comparison score is 1. Thus, we return the array $[1, 1]$.

Sample Input 1

```
17 28 30
99 16 8
```

Sample Output 1

```
2 1
```

Explanation 1

Comparing the 0^{th} elements, $17 < 99$ so Bob receives a point.
Comparing the 1^{st} and 2^{nd} elements, $28 > 16$ and $30 > 8$ so Alice receives two points.
The return array is $[2, 1]$.

Simple Array Sum

Given an array of N integers, can you find the sum of its elements?

Input Format

The first line contains an integer, N, denoting the size of the array.
The second line contains N space-separated integers representing the array's elements.

Output Format

Print the sum of the array's elements as a single integer.

Sample Input

```
6
1 2 3 4 10 11
```

Sample Output

```
31
```

Explanation

We print the sum of the array's elements, which is: $1 + 2 + 3 + 4 + 10 + 11 = 31$.

Staircase

Consider a staircase of size $n = 4$:

```
   #
  ##
 ###
####
```

Observe that its base and height are both equal to n, and the image is drawn using `#` symbols and spaces. *The last line is not preceded by any spaces.*

Write a program that prints a staircase of size n.

Input Format

A single integer, n, denoting the size of the staircase.

Output Format

Print a staircase of size n using `#` symbols and spaces.

Note: The last line must have 0 spaces in it.

Sample Input

```
6
```

Sample Output

```
     #
    ##
   ###
  ####
 #####
######
```

Explanation

The staircase is right-aligned, composed of `#` symbols and spaces, and has a height and width of $n = 6$.

Time Conversion

Given a time in 12-hour AM/PM format, convert it to military (24-hour) time.

Note: Midnight is 12:00:00AM on a 12-hour clock, and 00:00:00 on a 24-hour clock. Noon is 12:00:00PM on a 12-hour clock, and 12:00:00 on a 24-hour clock.

Input Format

A single string s containing a time in 12-hour clock format (i.e.: $\mathbf{hh:mm:ssAM}$ or $\mathbf{hh:mm:ssPM}$), where $01 \leq hh \leq 12$ and $00 \leq mm, ss \leq 59$.

Output Format

Convert and print the given time in 24-hour format, where $00 \leq hh \leq 23$.

Sample Input

```
07:05:45PM
```

Sample Output

```
19:05:45
```

Plus Minus

Given an array of integers, calculate the fractions of its elements that are *positive*, *negative*, and are *zeros*. Print the decimal value of each fraction on a new line.

Note: This challenge introduces precision problems. The test cases are scaled to six decimal places, though answers with absolute error of up to 10^{-4} are acceptable.

For example, given the array $arr = [1, 1, 0, -1, -1]$ there are 5 elements, two positive, two negative and one zero. Their ratios would be $\frac{2}{5} = 0.400000$, $\frac{2}{5} = 0.400000$ and $\frac{1}{5} = 0.200000$. It should be printed as

```
0.400000
0.400000
0.200000
```

Function Description

Complete the *plusMinus* function in the editor below. It should print out the ratio of positive, negative and zero items in the array, each on a separate line rounded to six decimals.

plusMinus has the following parameter(s):

- *arr*: an array of integers

Input Format

The first line contains an integer, n, denoting the size of the array.
The second line contains n space-separated integers describing an array of numbers $arr(arr[0], arr[1], arr[2], \ldots, arr[n-1])$.

Constraints

$0 < n \leq 100$
$-100 \leq arr[i] \leq 100$

Output Format

You must print the following 3 lines:

1. A decimal representing of the fraction of *positive* numbers in the array compared to its size.
2. A decimal representing of the fraction of *negative* numbers in the array compared to its size.
3. A decimal representing of the fraction of *zeros* in the array compared to its size.

Sample Input

```
6
-4 3 -9 0 4 1
```

Sample Output

```
0.500000
0.333333
0.166667
```

Explanation

There are 3 positive numbers, 2 negative numbers, and 1 zero in the array.

The proportions of occurrence are positive: $\frac{3}{6} = 0.500000$, negative: $\frac{2}{6} = 0.333333$ and zeros: $\frac{1}{6} = 0.166667$.

Diagonal Difference

Given a square matrix of size $N \times N$, calculate the absolute difference between the sums of its diagonals.

Input Format

The first line contains a single integer, N. The next N lines denote the matrix's rows, with each line containing N space-separated integers describing the columns.

Constraints

- $-100 \leq \text{Elements in the matrix} \leq 100$

Output Format

Print the absolute difference between the two sums of the matrix's diagonals as a single integer.

Sample Input

```
3
11 2 4
4 5 6
10 8 -12
```

Sample Output

```
15
```

Explanation

The primary diagonal is:

```
11
  5
    -12
```

Sum across the primary diagonal: 11 + 5 - 12 = 4

The secondary diagonal is:

```
    4
  5
10
```

Sum across the secondary diagonal: 4 + 5 + 10 = 19
Difference: |4 - 19| = 15

Note: |x| is absolute value function

Mini-Max Sum

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.

For example, $arr = [1, 3, 5, 7, 9]$. Our minimum sum is $1 + 3 + 5 + 7 = 16$ and our maximum sum is $3 + 5 + 7 + 9 = 24$. We would print

```
16 24
```

Function Description

Complete the *miniMaxSum* function in the editor below. It should print two space-separated integers on one line: the minimum sum and the maximum sum of 4 of 5 elements.

miniMaxSum has the following parameter(s):

- *arr*: an array of 5 integers

Input Format

A single line of five space-separated integers.

Constraints

$1 \leq arr[i] \leq 10^9$

Output Format

Print two space-separated long integers denoting the respective minimum and maximum values that can be calculated by summing exactly *four* of the five integers. (The output can be greater than a 32 bit integer.)

Sample Input

```
1 2 3 4 5
```

Sample Output

```
10 14
```

Explanation

Our initial numbers are 1, 2, 3, 4, and 5. We can calculate the following sums using four of the five integers:

1. If we sum everything except 1, our sum is $2 + 3 + 4 + 5 = 14$.
2. If we sum everything except 2, our sum is $1 + 3 + 4 + 5 = 13$.
3. If we sum everything except 3, our sum is $1 + 2 + 4 + 5 = 12$.
4. If we sum everything except 4, our sum is $1 + 2 + 3 + 5 = 11$.
5. If we sum everything except 5, our sum is $1 + 2 + 3 + 4 = 10$.

Hints: Beware of integer overflow! Use 64-bit Integer.

A Very Big Sum

You are given an array of integers of size N. You need to print the sum of the elements in the array, keeping in mind that some of those integers may be quite large.

Input Format

The first line of the input consists of an integer N. The next line contains N space-separated integers contained in the array.

Output Format

Print a single value equal to the sum of the elements in the array.

Constraints
$1 \leq N \leq 10$
$0 \leq A[i] \leq 10^{10}$

Sample Input

```
5
1000000001 1000000002 1000000003 1000000004 1000000005
```

Output

```
5000000015
```

Note:

The range of the 32-bit integer is $(-2^{31})\ to\ (2^{31}-1)\ or\ [-2147483648, 2147483647]$.

When we add several integer values, the resulting sum might exceed the above range. You might need to use long long int in C/C++ or long data type in Java to store such sums.