

Problem 1. Longest Distance

There are N hills, numbered from 0 to $N - 1$, arranged in a row. A couple of friends were standing on one hill when they had a terrible quarrel. They want to get away from one another so that the distance between them will be as large as possible. The distance between hills numbered J and K , where $J \leq K$, is computed as $K - J$. Friends can only go down, meaning that they can move from one hill to another only if the two hills are adjacent and the second hill is of the same or less height as the first.

Input:

first line – N (range 2 ~ 50)

second line - the heights of the N hills (range 1 ~ 100)

Output:

the longest possible distance that two friends can make between each other starting from one of the hills.

They can choose to stand on the optimal starting hill initially. In other words, each hill can be starting point.

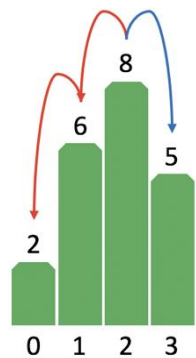
Examples:

1) Input:

4
2 6 8 5

Output:

3



Start
index #2

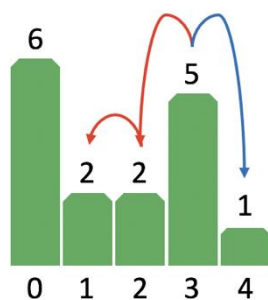
Distance
= 3 - 0
= 3

2) Input:

5
6 2 2 5 1

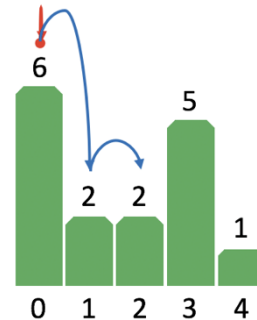
Output:

3



Start
index #3

Distance
= 4 - 1
= 3



Start
index #0

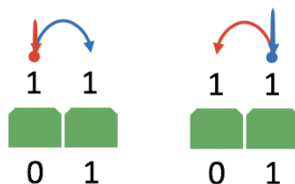
Distance
= 2 - 0
= 2

3) Input:

2
1 1

Output:

1



Start index #0 or #1

Distance
= 1 - 0 = 1 (same result)

Problem 2. A Very Simple Calculator

Very simple calculator, Calc, is a very simple calculator which can calculate 6 operators, variable assignment, print variable and exit. Make the this very simple calculator. (The program must print out the evaluation result. Please check the example output on Goorm.)

You can see the operators as below.

1. print : Print variable. Variable has ID a to z.

[input] a

[output] 7.5 # 7.5 is stored in variable 'a' before.

2. assign (=) : The result of add, sub, mul, div, rem or pow is stored in a variable. Unassigned value must be treated as zero.

[input] a = 15 / 2

[output] 7.5 # 7.5 will be stored in variable 'a'.

3. add (+) : Add two numbers(double type) or variables.

[input] 5 + 2.5

[output] 7.5

4. mul (*) : Multiply two numbers(double type) or variables

[input] a * 2

[output] 15

5. Sub (-) : Subtract two numbers(double type) or variables.

[input] 5 - 2.5

[output] 2.5

[input] a - 2

[output] 5.5

6. div (/) : Divide two numbers(double type) or variables.

7. rem (%) : Calculate remainder of two numbers or variables. In this case every value will be treated as integer.

[input] 7.5 % 2 # 7.5 will be converted to 7

[output] 1

[input] 7 % 2.5 # 2.5 will be converted to 2

[output] 1

8. pow (^) : Calculate the power of two numbers(double type) or variables.

[input] 2 ^ 2

[output] 4

[input] 7.5 ^ 2

[output] 56.25

9. exit : Exit the program

[input] exit

[output] # No output, just quit the program

* We only consider "valid" user inputs to make parser easily. For example,

1 + 1 => Valid input

1 + 1 + 1 => Invalid input. Consider two operands only.

10*10 => Valid input

10 => Invalid input. There are no cases consisting of only numbers.

Problem 3. Fraction Class

Make a new class named “Fraction” which represents fraction (분수) values in C++. Fraction class contains three private variables: N, D(Denominator, 분모), NU (Numerator분자) and it represents following fraction value.

```
int N;          // Integer
int D;          // Denominator
int NU;         // Numerator
```



$$N \frac{NU}{D}$$

Implements following public class functions in Fraction class. (Details about those functions are mentioned below)

Functions	Explanation
Fraction sum (Fraction b)	Add two fractions and return result.
Fraction sum (double b)	Add double and fraction value and return result.
Fraction multiply (Fraction b)	Multiply two fractions and return result
Fraction multiply (double b)	Multiply double and fraction value and return result.
void abbreviation ()	Abbreviate(약분) fraction.
bool toMixedNum ()	Convert fraction into mixed number. (Updates N, D and NU of instance) (if it is changed, return true. else return false.)
Void print ()	Print fraction value * Print format : N and NU/D
double toDouble ()	Convert fraction into double. (Round to fourth decimal place)

And makes two functions, **str2Fraction** which converts string into Fraction class, and **double2Fraction** which converts double into Fraction class. Details are mentioned below:

Functions	Explanation
Fraction str2Fraction (string str)	Converts string into Fraction class. * String format : N/NU/D
Fraction double2Fraction (double val)	Converts double value into Fraction class.

Input:

(fraction expressed in string – frac1) (double value – frac2)

Output:

Total 10 lines.

```
(frac1)
(frac2)
(frac1 + frac1)
(frac1 + given double value)
(frac2 * frac1)
(frac2 * given double value)
(frac1.toDouble)
(frac2.toDouble)
(frac1.toMixedNum)
(frac2.toMixedNum)
```

Examples:

Input	Output	Input	Output
	1 and 1/2		1 and 3/5
	1 and 1/4		1 and 1/160
	3 and 0/0		3 and 1/5
	2 and 3/4		2 and 97/160
1/1/2 1.25	1 and 7/8	1/3/5 1.00625	1 and 61/100
	1 and 9/16		1 and 321/25600
	1.500000		1.600000
	1.250000		1.006200
	1 and 1/2		1 and 3/5
	1 and 1/4		1 and 1/160

[Restriction]

1. All returned fractions should be abbreviated form. (
2. And all improper fractions(가분수) should be converted into mixed number form.
3. If there's no integer part in fraction, set N = 0.
4. If Denominator is 1, set D and NU to 0s and updates N.
5. You cannot use any other library except iostream and string.
6. **DO NOT MODIFY** setFraction function.

[Evaluation]

10 testcases, each 10 points total 100 points.