PracticeA - Welcome to AtCoder

Time Limit: 2 sec / Memory Limit: 256 MB

Problem

Your task is to process some data.

You are given 3 integers a , b , c and a string s. Output result of $a + b + c$ and string s with a half-width break.

Constrains

- $1 \leq a, b, c \leq 1,000$
- $1 \leq |s| \leq 100$

Input

Input will be given in the following format from Standard Input:

a
b c
s

Output

Output the result of $a + b + c$ and string s with a half-width break in one line.

Also, make sure to insert a line break at the end of the output.

Input Example #1

```
1
2 3
test
```

Output Example #1

```
6 test
```

- $1 + 2 + 3$ equals to 6.

Input Example #2

```
72
128 256
myonmyon
```

Output Example #2

```
456 myonmyon
```

- $72 + 128 + 256$ equals to 456.

Notice

If you are ' C ' or ' C++ ' programmer, please designate the types of $main$ function as ' int ' and not to forget ' return 0; '.

If you are ' Java ' programmer, please designate your class name as ' Main ', not ' main '.

References

We prepared some sample answers bellow (Not all programming languages).
Please use these examples as reference.

- C
- C++
- Java
- C#
- PHP
- D
- Go
- Python
- Perl
- Ruby
- Haskell
- Pascal
- JavaScript(Node.js)
- Scala

Example of C

```c
#include<stdio.h>

int main()
{
    int a,b,c;
    char s[101];
    // get a integer
    scanf("%d", &a);
    // get two integers separated half-width break
    scanf("%d %d",&b,&c);
    // get a string
    scanf("%s",s);
    // output
    printf("%d %s\n",a+b+c,s);
    return 0;
}
```

Example of C++

```cpp
#include<iostream>
using namespace std;

int main()
{
    // get a integer
    int a;
    cin >> a;
    // get two integers separated with half-width break
    int b,c;
    cin >> b >> c;
    // get a string
    string s;
    cin >> s;
    // output
    cout << (a+b+c) << " " << s << endl;
    return 0;
}
```

Example of Java

```java
import java.util.*;
public class Main {
        public static void main(String[] args){
                Scanner sc = new Scanner(System.in);
                // get a integer
                int a = sc.nextInt();
                // get two integers separated with half-width break
                int b = sc.nextInt();
                int c = sc.nextInt();
                // get a string
                String s = sc.next();
                // output
                System.out.println((a+b+c) + " " + s);
        }
}
```

Example of C#

```cs
using System;
class Program
{
        static void Main(string[] args)
        {
                // get a integer
                int a = int.Parse(Console.ReadLine());
                // get two integers separated with half-width break
                string[] input = Console.ReadLine().Split(' ');
                int b = int.Parse(input[0]);
                int c = int.Parse(input[1]);
                // get a string
                string s = Console.ReadLine();
                //output
                Console.WriteLine((a+b+c) + " " + s);
        }
}
```

Example of PHP

```php
<?php
# get a integer
fscanf(STDIN, "%d", $a);
# get two integers separated with half-width break
fscanf(STDIN, "%d %d", $b, $c);
# get a string
fscanf(STDIN, "%s", $s);
# output
echo ($a+$b+$c)." ".$s."\n";
?>
```

Example of D

```d
import std.stdio;
import std.string;
import std.conv;
void main()
{
        // get a integer
        int a = to!int(chomp(readln()));
        // get two integers separated with half-width break
        string[] input = split(readln());
        int b = to!int(input[0]);
        int c = to!int(input[1]);
        // get a string
        string s = chomp(readln());
        // output
        writefln("%d %s", a+b+c, s);
}
```

Example of Go

```go
package main

import (
        "fmt"
)

func main() {
        var a, b, c int
        var s string
        fmt.Scanf("%d", &a)
        fmt.Scanf("%d %d", &b, &c)
        fmt.Scanf("%s", &s)
        fmt.Printf("%d %s\n", a+b+c, s)
}
```

Example of Python

```python
# -*- coding: utf-8 -*-
# get a integer
a = int(raw_input())
# get two integers separated with half-width break
b, c = map(int, raw_input().split())
# get a string
s = raw_input()
# output
print str(a+b+c) + " " + s
```

Example of Perl

```perl
# get a integer
my $a = <STDIN>;
# get two integers separated with half-width break
my $input = <STDIN>;
chomp $input;
my ($b, $c) = split / /, $input;
$ret = $a + $b + $c;
# get a string
my $s = <STDIN>;
chomp $s;
# output
print "$ret $s\n";
```

Example of Ruby

```
# get a integer
a = gets.to_i
# get two integers separated with half-width break
b,c=gets.chomp.split(" ").map(&:to_i);
# get a string
s = gets.chomp
# output
print("#{a+b+c} #{s}\n")
```

Example of Haskell

```
{- supportedby @tanakh -}
import Control.Applicative
main :: IO ()
main = do
    -- get a integer
    a <- readLn
    -- get two integers separated with half-width break
    [b, c] <- map read . words <$> getLine
    -- get a string
    s <- getLine
    -- output
    putStrLn $ show (a + b + c) ++ " " ++ s
```

Example of Pascal

```
var
    a, b, c : integer;
    s : ShortString;
begin
    { get a integer }
    readln(a);
    { get two integers separated with half-width break }
    read(b);
    readln(c);
    { get a string }
    readln(s);
    { output }
    writeln(a+b+c, ' ', s);
end.
```

Example of JavaScript(Node.js)

```
// parameter "input" gets all data
function Main(input) {
        // the first line is assigned to input[0], the second line is assigned to input[1] similarly.
        input = input.split("\n");
        tmp = input[1].split(" ");
        // convert string from integer using "parseInt"
        var a = parseInt(input[0], 10);
        var b = parseInt(tmp[0], 10);
        var c = parseInt(tmp[1], 10);
        var s = input[2];
        //output
        console.log('%d %s',a+b+c,s);
}
// Don't edit this line!
Main(require("fs").readFileSync("/dev/stdin", "utf8"));
```

Example of Scala

```
// supported by @_Floojual
object Main extends App {
    var a = readInt
    var num = readLine
    var s = readLine
    var sum = a + num.split(" ")(0).toInt + num.split(" ")(1).toInt
    println(sum + " " + s);
}
```

ABC086A - Product

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 100 points

Problem Statement

AtCoDeer the deer found two positive integers, a and b. Determine whether the product of a and b is even or odd.

Constraints

- $1 \leq a, b \leq 10000$
- a and b are integers.

Input

Input is given from Standard Input in the following format:

```
a b
```

Output

If the product is odd, print `Odd`; if it is even, print `Even`.

Sample Input 1

```
3 4
```

Sample Output 1

```
Even
```

As $3 \times 4 = 12$ is even, print `Even`.

Sample Input 2

```
1 21
```

Sample Output 2

```
Odd
```

As $1 \times 21 = 21$ is odd, print `Odd`.

ABC081A - Placing Marbles

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 100 points

Problem Statement

Snuke has a grid consisting of three squares numbered $1, 2$ and 3. In each square, either `'0'` or `'1'` is written. The number written in Square i is s_i.

Snuke will place a marble on each square that says `'1'`. Find the number of squares on which Snuke will place a marble.

Constraints

- Each of s_1, s_2 and s_3 is either `'1'` or `'0'`.

Input

Input is given from Standard Input in the following format:

$s_1 s_2 s_3$

Output

Print the answer.

Sample Input 1

```
101
```

Sample Output 1

```
2
```

- A marble will be placed on Square 1 and 3.

Sample Input 2

```
000
```

Sample Output 2

```
0
```

- No marble will be placed on any square.

ABC081B - Shift only

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 200 points

Problem Statement

There are N positive integers written on a blackboard: $A_1, \ldots, A_N$.

Snuke can perform the following operation when all integers on the blackboard are even:

- Replace each integer X on the blackboard by X divided by 2.

Find the maximum possible number of operations that Snuke can perform.

Constraints

- $1 \leq N \leq 200$
- $1 \leq A_i \leq 10^9$

Input

Input is given from Standard Input in the following format:

N
$A_1 \quad A_2 \quad \ldots \quad A_N$

Output

Print the maximum possible number of operations that Snuke can perform.

Sample Input 1

```
3
8 12 40
```

Sample Output 1

```
2
```

Initially, $[8, 12, 40]$ are written on the blackboard. Since all those integers are even, Snuke can perform the operation.

After the operation is performed once, $[4, 6, 20]$ are written on the blackboard. Since all those integers are again even, he can perform the operation.

After the operation is performed twice, $[2, 3, 10]$ are written on the blackboard. Now, there is an odd number 3 on the blackboard, so he cannot perform the operation any more.

Thus, Snuke can perform the operation at most twice.

Sample Input 2

```
4
5 6 8 10
```

Sample Output 2

```
0
```

Since there is an odd number 5 on the blackboard already in the beginning, Snuke cannot perform the operation at all.

Sample Input 3

```
6
382253568 723152896 37802240 379425024 404894720 471526144
```

Sample Output 3

```
8
```

ABC087B - Coins

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 200 points

Problem Statement

You have A 500-yen coins, B 100-yen coins and C 50-yen coins (yen is the currency of Japan). In how many ways can we select some of these coins so that they are X yen in total?

Coins of the same kind cannot be distinguished. Two ways to select coins are distinguished when, for some kind of coin, the numbers of that coin are different.

Constraints

- $0 \leq A, B, C \leq 50$
- $A + B + C \geq 1$
- $50 \leq X \leq 20\,000$
- A, B and C are integers.
- X is a multiple of 50.

Input

Input is given from Standard Input in the following format:

$$
\begin{A}
A \\
B \\
C \\
X
\end{A}
$$

Output

Print the number of ways to select coins.

Sample Input 1

```
2
2
2
100
```

Sample Output 1

```
2
```

There are two ways to satisfy the condition:

- Select zero 500-yen coins, one 100-yen coin and zero 50-yen coins.
- Select zero 500-yen coins, zero 100-yen coins and two 50-yen coins.

Sample Input 2

```
5
1
0
150
```

Sample Output 2

```
0
```

Note that the total must be exactly X yen.

Sample Input 3

```
30
40
50
6000
```

Sample Output 3

```
213
```

ABC083B - Some Sums

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 200 points

Problem Statement

Find the sum of the integers between 1 and N (inclusive), whose sum of digits written in base 10 is between A and B (inclusive).

Constraints

- $1 \leq N \leq 10^4$
- $1 \leq A \leq B \leq 36$
- All input values are integers.

Input

Input is given from Standard Input in the following format:

```
N A B
```

Output

Print the sum of the integers between 1 and N (inclusive), whose sum of digits written in base 10 is between A and B (inclusive).

Sample Input 1

```
20 2 5
```

Sample Output 1

```
84
```

Among the integers not greater than 20, the ones whose sums of digits are between 2 and 5, are:
$2, 3, 4, 5, 11, 12, 13, 14$ and 20. We should print the sum of these, 84.

Sample Input 2

```
10 1 2
```

Sample Output 2

```
13
```

Sample Input 3

```
100 4 16
```

Sample Output 3

```
4554
```

ABC088B - Card Game for Two

Time Limit: 2 sec / Memory Limit: 256 MB

Score: 200 points

Problem Statement

We have N cards. A number a_i is written on the i-th card.
Alice and Bob will play a game using these cards. In this game, Alice and Bob alternately take one card. Alice goes first. The game ends when all the cards are taken by the two players, and the score of each player is the sum of the numbers written on the cards he/she has taken. When both players take the optimal strategy to maximize their scores, find Alice's score minus Bob's score.

Constraints

- N is an integer between 1 and 100 (inclusive).
- a_i $(1 \leq i \leq N)$ is an integer between 1 and 100 (inclusive).

Input

Input is given from Standard Input in the following format:

N
a_1 a_2 a_3 $\ldots$ a_N

Output

Print Alice's score minus Bob's score when both players take the optimal strategy to maximize their scores.

Sample Input 1

```
2
3 1
```

Sample Output 1

```
2
```

First, Alice will take the card with 3. Then, Bob will take the card with 1. The difference of their scores will be 3 - 1 = 2.

Sample Input 2

```
3
2 7 4
```

Sample Output 2

```
5
```

First, Alice will take the card with 7. Then, Bob will take the card with 4. Lastly, Alice will take the card with 2. The difference of their scores will be 7 - 4 + 2 = 5. The difference of their scores will be 3 - 1 = 2.

Sample Input 3

```
4
20 18 2 18
```

Sample Output 3

```
18
```

ABC085B - Kagami Mochi

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 200 points

Problem Statement

An *X-layered kagami mochi* $(X \geq 1)$ is a pile of X round mochi (rice cake) stacked vertically where each mochi (except the bottom one) has a smaller diameter than that of the mochi directly below it. For example, if you stack three mochi with diameters of 10, 8 and 6 centimeters from bottom to top in this order, you have a 3-layered kagami mochi; if you put just one mochi, you have a 1-layered kagami mochi.

Lunlun the dachshund has N round mochi, and the diameter of the i-th mochi is d_i centimeters. When we make a kagami mochi using some or all of them, at most how many layers can our kagami mochi have?

Constraints

- $1 \leq N \leq 100$
- $1 \leq d_i \leq 100$
- All input values are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{aligned}
&N\\
&d_1\\
&:\\
&d_N
\end{aligned}
$$

Output

Print the maximum number of layers in a kagami mochi that can be made.

Sample Input 1

```
4
10
8
8
6
```

Sample Output 1

```
3
```

If we stack the mochi with diameters of 10, 8 and 6 centimeters from bottom to top in this order, we have a 3-layered kagami mochi, which is the maximum number of layers.

Sample Input 2

```
3
15
15
15
```

Sample Output 2

```
1
```

When all the mochi have the same diameter, we can only have a 1-layered kagami mochi.

Sample Input 3

```
7
50
30
50
100
50
80
30
```

Sample Output 3

```
4
```

ABC085C - Otoshidama

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 300 points

Problem Statement

The commonly used bills in Japan are 10000-yen, 5000-yen and 1000-yen bills. Below, the word "bill" refers to only these.

According to Aohashi, he received an otoshidama (New Year money gift) envelope from his grandfather that contained N bills for a total of Y yen, but he may be lying. Determine whether such a situation is possible, and if it is, find a possible set of bills contained in the envelope. Assume that his grandfather is rich enough, and the envelope was large enough.

Constraints

- $1 \leq N \leq 2000$
- $1000 \leq Y \leq 2 \times 10^7$
- N is an integer.
- Y is a multiple of 1000.

Input

Input is given from Standard Input in the following format:

```
N Y
```

Output

If the total value of N bills cannot be Y yen, print `-1 -1 -1`.

If the total value of N bills can be Y yen, let one such set of bills be "x 10000-yen bills, y 5000-yen bills and z 1000-yen bills", and print x, y, z with spaces in between. If there are multiple possibilities, any of them may be printed.

Sample Input 1

```
9 45000
```

Sample Output 1

```
4 0 5
```

If the envelope contained 4 10000-yen bills and 5 1000-yen bills, he had 9 bills and 45000 yen in total. It is also possible that the envelope contained 9 5000-yen bills, so the output `0 9 0` is also correct.

Sample Input 2

```
20 196000
```

Sample Output 2

```
-1 -1 -1
```

When the envelope contained 20 bills in total, the total value would be 200000 yen if all the bills were 10000-yen bills, and would be at most 195000 yen otherwise, so it would never be 196000 yen.

Sample Input 3

```
1000 1234000
```

Sample Output 3

```
14 27 959
```

There are also many other possibilities.

Sample Input 4

```
2000 20000000
```

Sample Output 4

```
2000 0 0
```

ABC049C - 白昼夢 / Daydream

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 300 points

Problem Statement

You are given a string S consisting of lowercase English letters. Another string T is initially empty. Determine whether it is possible to obtain $S = T$ by performing the following operation an arbitrary number of times:

- Append one of the following at the end of T: `dream`, `dreamer`, `erase` and `eraser`.

Constraints

- $1 \leqq |S| \leqq 10^5$
- S consists of lowercase English letters.

Input

The input is given from Standard Input in the following format:

```
S
```

Output

If it is possible to obtain $S = T$, print `YES`. Otherwise, print `NO`.

Sample Input 1

```
erasedream
```

Sample Output 1

```
YES
```

Append `erase` and `dream` at the end of T in this order, to obtain $S = T$.

Sample Input 2

```
dreameraser
```

Sample Output 2

```
YES
```

Append ' `dream` ' and ' `eraser` ' at the end of T in this order, to obtain $S = T$.

Sample Input 3

```
dreamerer
```

Sample Output 3

```
NO
```

ABC086C - Traveling

Time Limit: 2 sec / Memory Limit: 256 MB

Score : 300 points

Problem Statement

AtCoDeer the deer is going on a trip in a two-dimensional plane. In his plan, he will depart from point $(0, 0)$ at time 0, then for each i between 1 and N (inclusive), he will visit point (x_i, y_i) at time t_i.

If AtCoDeer is at point (x, y) at time t, he can be at one of the following points at time $t + 1$: $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ and $(x, y - 1)$. Note that **he cannot stay at his place**. Determine whether he can carry out his plan.

Constraints

- $1 \leq N \leq 10^5$
- $0 \leq x_i \leq 10^5$
- $0 \leq y_i \leq 10^5$
- $1 \leq t_i \leq 10^5$
- $t_i < t_{i+1}$ $(1 \leq i \leq N - 1)$
- All input values are integers.

Input

Input is given from Standard Input in the following format:

$$
\begin{array}{lll}
N & & \\
t_1 & x_1 & y_1 \\
t_2 & x_2 & y_2 \\
: & & \\
t_N & x_N & y_N
\end{array}
$$

Output

If AtCoDeer can carry out his plan, print `'Yes'`; if he cannot, print `'No'`.

Sample Input 1

```
2
3 1 2
6 1 1
```

Sample Output 1

```
Yes
```

For example, he can travel as follows: $(0, 0)$, $(0, 1)$, $(1, 1)$, $(1, 2)$, $(1, 1)$, $(1, 0)$, then $(1, 1)$.

Sample Input 2

```
1
2 100 100
```

Sample Output 2

```
No
```

It is impossible to be at $(100, 100)$ two seconds after being at $(0, 0)$.

Sample Input 3

```
2
5 1 1
100 1 1
```

Sample Output 3

```
No
```