2022 KAIST RUN Spring Contest

Sponsored By:

MOLOCO

NAVER D2

DEVSISTERS

Samsung Software Membership

STARTLINK

Rules

- This contest is 2022 KAIST RUN Spring Contest.
- This contest is sponsored by MOLOCO, Naver D2, and DEVSISTERS, Samsung Software Membership, and Startlink.
- You can only participate individually.
- Use of the network is prohibited during the competition, except for submitting source codes and accessing language reference sites, and using translation. Here are the allowed reference and translation sites.
  - `C/C++           : https://en.cppreference.com/w/`
  - `Java            : https://docs.oracle.com/javase/8/docs/api/`
  - `Python          : https://docs.python.org/`
  - `Kotlin          : https://kotlinlang.org/docs/reference/`
  - `Naver Papago    : https://papago.naver.com/`
  - `Google Translate: https://translate.google.com/`
- Use of pre-written code or template, any library is prohibited. You should only submit code written by yourself during contest time.
- The contest lasts 4 hours.
- The contest consists of 8 problems.
- Problems are ordered roughly by the difficulty of getting full points. You are strongly encouraged to read all the problems.
- Each problem consists of 1 or more subtasks, and each subtask is worth a certain number of points. Subtasks are easier versions of the problems with more restrictive constraints.
- Each subtask consists of several test cases, and points for the subtask are awarded if every test case is passed. The total number of points for each problem is 100 points.
- Each problem has a time limit and memory limit. This means your problem should run in the given time and memory limit for each test case.
- Every problem is guaranteed to be solvable using `C++17`. This is not guaranteed for any other language.
- The memory limit for every problem is 1024MiB.
- Each participant's ranking is determined in the following way.
  - Penalty time = (Sum of duration from contest start to the last submission that increased points per each problem)
  - Ranking = (# of participants with higher points) + (# of participants with same points and lower penalty time) + 1

Language Guide

- You can choose your programming language from among the following:

```
C11   :  gcc (Ubuntu 10.2.0-5ubuntu1~20.04) 10.2.0
C++17 :  g++ (Ubuntu 10.2.0-5ubuntu1~20.04) 10.2.0
Java  :  OpenJDK Runtime Environment (build 11.0.11+9-Ubuntu-0ubuntu2.20.04)
Kotlin:  kotlinc-jvm 1.5.0 (JRE 11.0.11+9-Ubuntu-0ubuntu2.20.04)
PyPy 3:  Python 3.7.10 (7.3.4+dfsg-1~ppa1~ubuntu20.04, Apr 12 2021, 21:32:29) [PyPy
         7.3.4 with GCC 9.3.0]
```

- In `Java`, your class name which includes the `main` method should be `Main`.
- Compilation commands are available at the contest system.

Sample Code

- Followings are sample codes which reads two space-separated integers from standard input, and prints their sum to standard output.

  - C11 / C++17

    ```
    #include <stdio.h>

    int main() {
        int a, b;
        scanf("%d%d", &a, &b);
        printf("%d\n", a+b);
        return 0;
    }
    ```

  - Java

    ```
    import java.util.*;

    public class Main{
        public static void main(String args[]){
            Scanner sc = new Scanner(System.in);
            int a, b;
            a = sc.nextInt();
            b = sc.nextInt();
            System.out.println(a + b);
        }
    }
    ```

  - Kotlin

    ```
    import java.util.Scanner

    fun main(args: Array<String>) {
        val sc: Scanner = Scanner(System.‘in‘)
        var a = sc.nextInt()
        var b = sc.nextInt()
        println(a+b)
    }
    ```

  - PyPy 3

    ```
    import sys

    def main():
        a, b = map(int, sys.stdin.readline().split())
        print(a+b)

    if __name__ == ’__main__’:
        main()
    ```

Problem list

#	Problem Name	Time limit	Score	Subtask score					
		(All languages)	Full score	1	2	3	4	5	6
A	Sequnce Conversion	3 seconds	100	9	91				
B	Ontongdaejeon	1 second	100	14	26	60			
C	TOO EASY Cookie Run	2 seconds	100	10	20	30	40		
D	Sequence Conversion 2	2 seconds	100	20	30	50			
E	Comparing Fractions	3 seconds	100	5	13	21	22	18	21
F	Tree GCD	2.5 seconds	100	10	20	70			
G	Counting Rectangles	5 seconds	100	30	15	55			
H	Strange Graph	7 seconds	100	7	23	46	24		

Problem A. Sequence conversion

Time limit: 3 seconds

You are given two arrays of non-negative integers $a_1, a_2, \ldots, a_N$ and $b_1, b_2, \ldots, b_N$.

You can perform the following operation several times:

- Choose a non-negative integer x and an index $1 \le i < N$. Then, change a_i to $a_i \oplus x$ and change a_{i+1} to $a_{i+1} \oplus x$.

Expression $x \oplus y$ means **bitwise xor** of two numbers x and y. In binary representation, if the i-th digit of x and y is equal, then the i-th digit of $x \oplus y$ is 0, and if not, it is 1. The given operation exists in all modern programming languages. For example, in *C++* and *Java*, it is represented as x ^ y.

You want to convert $\{a_i\}$ to $\{b_i\}$ by performing the minimum number of operations.

Find the minimum number of operations to convert $\{a_i\}$ to $\{b_i\}$.

If you cannot convert $\{a_i\}$ to $\{b_i\}$ with the given operation, print -1.

Input

The first line contains an integer N, where N denotes the length of the two sequences.

The second line contains N space-separated non-negative integers $a_1, a_2, \ldots, a_N$.

The third line contains N space-separated non-negative integers $b_1, b_2, \ldots, b_N$

Output

Print -1, if it is impossible to change the sequence $\{a_i\}$ to $\{b_i\}$.
Otherwise, print the minimum number of operations needed to change the sequence $\{a_i\}$ to $\{b_i\}$.

Constraints

- $1 \le N \le 10^6$
- $0 \le a_i, b_i < 2^{30}$ $(1 \le i \le N)$

Subtask 1 (9 points)

This subtask has an additional constraint:

- $N \le 5$

Subtask 2 (91 points)

This subtask has no additional constraints.

Examples

standard input	standard output

3
1 2 3
3 2 1
	2

3
1 5 3
1 2 3
	-1

Problem B. Ontongdaejeon

Time limit: 1 second

You have to buy N items to prepare RUN Spring Contest.

In Daejeon, there is a famous local currency, *Ontongdaejeon*. Using Ontongdaejeon, you can get a cashback point when purchasing an item – precisely **10 percent of the cash** you have paid as a **cashback point**.

If you have already got some cashback point from the previous purchases, you can use it instead of any integer amount of cash you want.

Specifically, you can use the cashback point with the following steps:

1. Let c be the amount of cashback point you already have and p be the price of an item you want to purchase.
2. You should determine the amount of cashback point to use. Let this be x.
3. x should satisfy following conditions:
    - $x \le p$
    - $x \le c$
    - x must be a non-negative integer.
4. Then, you can use x cashback points and $p - x$ cash to buy the item.
5. After buying the item, the cashback point you would have is $(c - x) + \frac{p - x}{10}$.

Find the minimum amount of cash required to buy all N items in a given order.

Before buying the first item, you have no cashback points.

Input

The first line contains one integer N, where N denotes the number of items.

The second line contains N space-separated integers $P_1, P_2, \ldots, P_N$ where P_i denotes the price of i-th item.

Output

Find the minimum amount of cash required to buy all N items in a given order using Ontongdaejeon.

Constraints

- $1 \le N \le 200\,000$
- $1 \le P_i \le 10^9$ $(1 \le i \le N)$

Subtask 1 (14 points)

This subtask has an additional constraint:

- $N \le 2$

Subtask 2 (26 points)

This subtask has additional constraints:

- $N \leq 100$
- $P_i \leq 100$ $(1 \leq i \leq N)$

Subtask 3 (60 points)

This subtask has no additional constraint.

Examples

standard input	standard output
3 100 10 12	111
1 100	100

Problem C. TOO EASY Cookie Run

Time limit: 2 seconds

Cookie Run game is a popular game in which the cookie character runs through a map consisting of N stages to score points.

Soo Young, the map designer for Cookie Run, was preparing a new map patch for Children's Day. After working hard to create an attractive map and testing it for the last time the day before the patch, she suddenly realized that the map was designed to be **too easy**!
The map design of the cookie run game consists of the following rules.

- Each stage is given a difficulty level of $A_0, A_1, A_2, ..., A_{N-1}$, and the higher the difficulty is, the more difficult it is to pass the corresponding stage.
- If the sum of the difficulty levels of consecutive stages is greater than or equal to M, we call this section as **interesting section**. That is, if $A_i + A_{i+1} + ... + A_j \geq M$, section (i, j) is an interesting section.
- Maps should always have at least K interesting sections.

In other words, the following conditions should be satisfied:

- Let T be the number of pairs (i, j) satisfying the following conditions:
  - $0 \leq i \leq j \leq N - 1$
  - $A_i + A_{i+1} + ... + A_j \geq M$.
- Then, $T \geq K$ for the given integer K.

To solve this problem, Soo Young requested help from KAIST RUN Spring Contest participants.
Since there is not much time left until the patch release, all Soo Young can do is **add difficulty X to all stages at once** to make it more difficult.
Your job is to find the smallest non-negative integer X that satisfies these conditions.

Input

The first line contains three space-separated integers, N, M, and K.
The second line contains N space-separated integers $A_0, A_1, \cdots, A_{N-1}$.

Output

Print a single non-negative integer denoting the smallest possible X.
You can assume that at least one non-negative integer X exists, satisfying the given conditions.

Constraints

- $1 \le N \le 100\,000$
- $1 \le M \le 10^{18}$
- $1 \le K \le \frac{N(N+1)}{2}$
- $0 \le A_i \le 10^9$ $(0 \le i \le N-1)$

Subtask 1 (10 points)

This subtask has additional constraints:

- Smallest possible $X \le 100$
- $N \le 1\,000$

Subtask 2 (20 points)

This subtask has additional constraints:

- Smallest possible $X \le 100$
- $N \le 30\,000$

Subtask 3 (30 points)

This subtask has an additional constraint:

- $N \le 5\,000$

Subtask 4 (40 points)

This subtask has no additional constraints.

Examples

standard input	standard output
3 20 5 4 0 4	16
5 30 4 1 2 3 4 5	6
4 32 4 8 7 5 9	9

Problem D. Sequence Conversion 2

Time limit: 2 second

You are given an array of non-negative integers $a_1, a_2, \ldots, a_N$.

You can perform the following operation several times:

- Choose an index i. ($1 \le i < \mathit{length\ of\ the\ array}$) Then, remove a_i, a_{i+1} and replace them with $a_i \oplus a_{i+1}$. (The total length of the array decreases by 1)

Expression $x \oplus y$ means **bitwise xor** of two numbers x and y. In binary representation, if the i-th digit of x and y is equal, then the i-th digit of $x \oplus y$ is 0, and if not, it is 1. The given operation exists in all modern programming languages. For example, in *C++* and *Java*, it is represented as x ^ y.

You want to convert the given array into **zig-zag** array.

We say an array of m integers, $z_1, z_2, ..., z_m$, is a zig-zag array if no three consecutive elements in the array are either monotonically increasing or monotonically decreasing.

In other words, if there are three elements z_i, z_{i+1}, z_{i+2} in the array such that $z_i \le z_{i+1} \le z_{i+2}$ or $z_i \ge z_{i+1} \ge z_{i+2}$ holds, the array is **not** zig-zag. Otherwise, it is zig-zag array.

Find the minimum number of operations needed to convert $\{a_i\}$ into a **zig-zag** array.

Input

The first line contains an integer N, where N denotes the length of the sequence.

The second line contains N space-separated non-negative integers $a_1, a_2, \ldots, a_N$.

Output

Print the minimum number of operations needed to convert the sequence $\{a_i\}$ into a zig-zag array.

Constraints

- $1 \le n \le 3\,000$
- $0 \le a_i < 2^{30}$ $(1 \le i \le N)$

Subtask 1 (20 points)

This subtask has an additional constraint:

- $N \le 20$

Subtask 2 (30 points)

This subtask has an additional constraint:

- $N \le 100$

Subtask 3 (50 points)

This subtask has no additional constraints.

Examples

standard input	standard output
3 1 2 3	1
3 1 3 2	0

Problem E. Comparing Fractions

Time limit: 3 seconds

This problem is interactive. Refer to the Interaction section below for a better understanding.

Today was the first class of CS999. First, you have learned non-negative integer less than or equal to 4×10^{18}. You have also learned the addition, subtraction, and comparison of two integers. The homework is to compare two fractions $\frac{A}{B}$ and $\frac{C}{D}$. Professor told you that you can solve homework only using classroom materials.

You are only allowed to use the following operations:

- **Addition**: For two integers a and b, you can calculate $a + b$. The result should be less than or equal to 4×10^{18}.
- **Subtraction**: For two integers a and b, you can calculate $a - b$. The result should be non-negative.
- **Comparison**: For two integers a and b, you can know whether two elements are equal or which one is greater.

By using these operations, you have to compare two fractions.

Interaction Protocol

You cannot get A, B, C, nor D directly in this problem. Instead, you can use some operations to a hidden array of length 10^6; $X_1, X_2, \cdots, X_{10^6}$. Initially, X satisfies $X_1 = A$, $X_2 = B$, $X_3 = C$, $X_4 = D$, and $X_i = 0$ for i greater than 4.

You can use the following commands to do operations:

- To add two elements of X, print a string "`+` i j k" ($1 \le i, j, k \le 10^6$). X_i will be replaced with a value of $X_j + X_k$. To perform this operation, $X_j + X_k$ must be less than or equal to 4×10^{18}. If not, you will receive a verdict of "Wrong Answer." Nothing will be given to the input.
- To subtract two elements of X, print a string "`-` i j k" ($1 \le i, j, k \le 10^6$). X_i will be replaced with a value of $X_j - X_k$. To perform this operation, $X_j \ge X_k$ must be satisfied. If not, you will receive a verdict of "Wrong Answer." Nothing will be given to the input.
- To compare two elements of X, print a string "`<` i j" ($1 \le i, j \le 10^6$). A line containing one integer will be given to the input.
  - "-1" will be given when $X_i < X_j$.
  - "0" will be given when $X_i = X_j$.
  - "1" will be given when $X_i > X_j$.

After successfully comparing $\frac{A}{B}$ and $\frac{C}{D}$, answer the comparison result by

- if $\frac{A}{B} < \frac{C}{D}$, print "`!` -1";
- if $\frac{A}{B} = \frac{C}{D}$, print "`!` 0";
- if $\frac{A}{B} > \frac{C}{D}$, print "`!` 1";

and terminate your program immediately.

If you make an invalid query, the interactor will terminate immediately and your program will receive a verdict of "Wrong Answer."

After printing each line, do not forget to flush the output. You can flush your output by

- `fflush(stdout);` in C11/C++17;
- `System.out.flush();` in Java;
- `System.out.flush()` in Kotlin;
- `sys.stdout.flush()` in PyPy3.

Constraints

- $1 \le A, B, C, D \le 10^9$

Scoring

If your comparison is incorrect, your score will be 0.

Otherwise, the score of program will be graded with three factors:

- i_{max}, the maximum index of array X you have used in operations.
- n_{op}, the number of operations you have done. The answer command will not be counted.
- X_{max}, the maximum number in the array X during operations.

Subtask 1 (5 points)

This subtask has an additional constraint:

- $\{A, B, C, D\} = \{1, 2, 3, 4\}$

Your program should satisfy:

- $n_{op} \le 20\,000$

Subtask 2 (13 points)

This subtask has an additional constraint:

- $A, B, C, D \le 10\,000$

Your program should satisfy:

- $i_{max} \le 100$
- $n_{op} \le 20\,000$
- $X_{max} \le 10^9$

Subtask 3 (21 points)

This subtask has an additional constraint:

- $A, B, C, D \leq 10^6$

Your program should satisfy:

- $i_{max} \leq 100$
- $n_{op} \leq 10\,000$
- $X_{max} \leq 2 \times 10^9$

Subtask 4 (22 points)

This subtask has no additional constraint.

Your program should satisfy:

- $i_{max} \leq 100$
- $n_{op} \leq 1\,000$

Subtask 5 (18 points)

This subtask has no additional constraint.

Your program should satisfy:

- $i_{max} \leq 100$
- $n_{op} \leq 1\,000$
- $X_{max} \leq 2 \times 10^9$

Subtask 6 (21 points)

This subtask has no additional constraint.

Your program should satisfy:

- $i_{max} \leq 100$
- $n_{op} \leq 1\,000$
- $X_{max} \leq 10^9$

Examples

standard input	standard output
-1	+ 5 3 1 - 1 2 1 < 1 5 ! 0
1 -1 0	< 1 2 < 4 3 < 2 2 ! -1

- For the first example, $A = 1$, $B = 1$, $C = 2$, $D = 2$. It satisfies the constraint of subtask 2, 3, 4, and 5.
- For the second example, $A = 1$, $B = 2$, $C = 3$, $D = 4$. It satisfies the constraint of every subtask.

Problem F. Tree GCD

Time limit: 2.5 seconds

You are given an undirected tree with N vertices, labeled with distinct integers from 1 to N.

Let's denote $\mathrm{dist}(i, j)$ be the length of the shortest path between two vertices i and j on the tree.

Find $\sum_{1 \le i < j \le N} \gcd(i, j, \mathrm{dist}(i, j))$. $\gcd(a, b, c)$ means the greatest common divisor of a, b, and c.

Input

The first line contains an integer N.

Each of the following $N - 1$ lines contains two space-separated integers u_i and v_i $(1 \le i \le N - 1)$, which means that there is an edge between them.

Output

Print the value of $\sum_{1 \le i < j \le N} \gcd(i, j, \mathrm{dist}(i, j))$.

Constraints

- $3 \le N \le 100\,000$
- $1 \le u_i \le N$ $(1 \le i \le N - 1)$
- $1 \le v_i \le N$ $(1 \le i \le N - 1)$
- The given graph is a tree.

Subtask 1 (10 points)

This subtask has an additional constraint:

- $N \le 5\,000$

Subtask 2 (20 points)

This subtask has an additional constraint:

- The degree of each vertex is less than 3.

Subtask 3 (70 points)

This subtask has no additional constraints.

Examples

standard input	standard output

3
1 2
1 3
	3

4
1 2
1 3
1 4
	7

6
1 3
1 5
5 2
6 5
5 4
	20

Notes

In the first example, $\gcd(i,\, j,\, \mathrm{dist}(i,\, j)) = 1$ for all $1 \le i < j \le 3$.
Thus, $\sum_{1 \le i < j \le 3} \gcd(i,\, j,\, \mathrm{dist}(i,\, j)) = 1 + 1 + 1 = 3$.

In the second example, $\gcd(i,\, j,\, \mathrm{dist}(i,\, j)) = 1$ for all $1 \le i < j \le 4$, except the case where $\gcd(2,\, 4,\, \mathrm{dist}(2,\, 4)) = 2$. Thus, $\sum_{1 \le i < j \le 4} \gcd(i,\, j,\, \mathrm{dist}(i,\, j)) = 1 + 1 + 1 + 1 + 1 + 2 = 7$.

Problem G. Counting Rectangles

Time limit: 5 seconds

For two arrays of integers A of size N and B of size M, we define a grid $G(A, B)$ of size $N \times M$, where cell (i, j) is colored black if $0 \le A_i + B_j$ and white otherwise.

We also define $F(A, B)$ as the number of black rectangles inside $G(A, B)$, where each cell of $G(A, B)$ is either entirely included in or disjoint with the rectangle.

In other words, $F(A, B)$ is equal to the number of tuples (l_1, r_1, l_2, r_2) such that $1 \le l_1 \le r_1 \le N$, $1 \le l_2 \le r_2 \le M$ and each cell (i, j) in $G(A, B)$ is colored black for all i, j such that $l_1 \le i \le r_1$, $l_2 \le j \le r_2$.

Initially, only A_1 and B_1 are given.

Then, you should process following Q queries:

- 0 v: append v to current array A.
- 1 v: append v to current array A. Then, print $F(A, B) \bmod 998\,244\,353$.
- 2 v: append v to current array B.
- 3 v: append v to current array B. Then, print $F(A, B) \bmod 998\,244\,353$.

Input

The first line contains one integer Q.

The second line contains two space-separated integers, A_1 and B_1.

Each of the following Q lines contains two space-separated integers denoting the queries in the described form.

Output

For each query of types 1 and 3, output a single integer denoting the answer to that query. Each answer should go on its own line.

Constraints

Let N be the size of array A after processing all queries, and M be the size of array B after processing all queries.

- $1 \le N \le 250\,000$
- $1 \le M \le 250\,000$
- $1 \le Q = N + M - 2$
- $-10^9 \le A_i \le 10^9$ $(1 \le i \le N)$
- $-10^9 \le B_i \le 10^9$ $(1 \le i \le M)$
- The last query has a type of 1 or 3.

Subtask 1 (30 points)

This subtask has additional constraints:

- $N \leq 2\,500$
- $M \leq 2\,500$

Subtask 2 (15 points)

This subtask has an additional constraint:

- Every query except the last one has a type of 0 or 2.

Subtask 3 (55 points)

This subtask has no additional constraints.

Examples

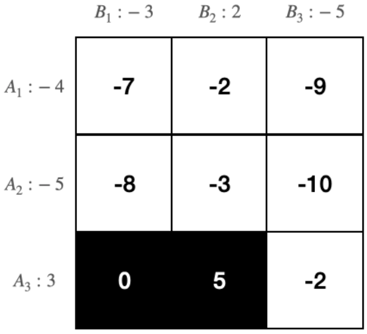
standard input	standard output

4
-4 -3
0 -5
2 2
0 3
3 -5
	3

8
-187121777 648583176
0 536185451
1 77324177
2 -543947071
1 -495948203
2 809620127
2 918209957
3 -724806401
1 30094601
	6
	10
	40
	60

Note

Grid $G(A, B)$ made from the first example is represented in the following picture. The number inside a cell $(i,\, j)$ denotes $A_i + B_j$.



	$B_1 : -3$	$B_2 : 2$	$B_3 : -5$
$A_1 : -4$	-7	-2	-9
$A_2 : -5$	-8	-3	-10
$A_3 : 3$	0	5	-2

Problem H. Strange Graph

Time limit: 7 seconds

You are given a two-dimensional array of integers of size $N \times K$, $A = A_{0,0}, A_{0,1}, \ldots, A_{0,K-1}, A_{1,0}, \ldots, A_{N-1,K-1}$ and also arrays of integers of size M, $U = U_0, \ldots, U_{M-1}$ and $V = V_0, \ldots, V_{M-1}$.

Jimin made a cute weighted undirected graph G, which is a complete graph with the weight of an edge connecting vertices u and v is $\left|A_{u,(v \bmod K)} - A_{v,(u \bmod K)}\right|$. Eunsoo then found the minimum spanning tree of G.

However, Jongyoung brutally deleted edges of G connecting U_i and V_i for $0 \le i \le M-1$. Note that G may not be connected after deleting the edges.

Now, to help poor Jimin and Eunsoo, you should find the minimum spanning forest of G. A minimum spanning forest is a union of the minimum spanning trees of its connected components.

Input

The first line contains three space-separated integers, N, K, and M.

Each of the following N lines contains K space-separated integers, $A_{i,0}, \ldots, A_{i,K-1}$. $(0 \le i \le N-1)$

Each of the following M lines contains two space-separated integers, U_i and V_i. $(0 \le i \le M-1)$

Output

Output the sum of the weight of edges in the minimum spanning forest of G.

Constraints

- $1 \le NK \le 300\,000$
- $1 \le K \le N$
- $0 \le M \le \min\left(\frac{N(N-1)}{2},\ 300\,000\right)$
- $-10^9 \le A_{i,j} \le 10^9$ $(0 \le i \le N-1, 0 \le j \le K-1)$
- $0 \le U_i < V_i \le N-1$ $(0 \le i \le M-1)$
- $(U_i, V_i) \ne (U_j, V_j)$ $(0 \le i < j \le M-1)$

Subtask 1 (7 points)

This subtask has an additional constraint:

- $N \le 1\,000$

Subtask 2 (23 points)

This subtask has additional constraints:

- $NK \le 150\,000$
- $M = 0$

Subtask 3 (46 points)

This subtask has additional constraints:

- $NK \leq 150\,000$
- $M \leq 150\,000$

Subtask 4 (24 points)

This subtask has no additional constraints.

Examples

standard input	standard output

7 1 7
0
1
2
1
2
-5
-1
4 5
1 3
4 6
0 4
2 5
1 4
3 4
	8

7 2 7
5 1
4 5
-2 4
4 1
-5 -5
2 -1
3 3
5 6
0 5
0 3
1 2
4 6
2 3
2 6
	11