BIUPC-2025 Spring Preliminary Contest (Junior Division)

A. Hello World ?

1 second, 256 megabytes

Author : Tiny Coder

Intake : 49

Dept : CSE

Just Print "hello wolrd"

Input
No Input. I mean what is the use for input?

Output
Only line of the output is "hello wolrd"

input
No Input is Given (This is not the input)
output
hello wolrd

B. Maximum Gcd

1 second, 256 megabytes

You are given an array of n positive integers.

Your task is to choose exactly k elements from the array such that the greatest common divisor (GCD) of the chosen elements is as large as possible.

Print the maximum GCD that can be achieved by choosing exactly k elements from the array.

Input
The first line contains two integers n and k — the size of the array and the number of elements to choose. ($1 \le k \le n \le 2 \times 10^5$)

The second line contains n integers $a_1, a_2, ..., a_n$ — the elements of the array. ($1 \le a_i \le 10^6$)

Output
Print a single integer — the maximum GCD that can be achieved by selecting exactly k elements from the array.

input
6 3 1 2 3 4 5 6
output
2

C. Shukrobar Chutir Din

1 second, 256 megabytes

Thanos, the ex-Avengers villain turned farmer, owner of K cows. Each cow produces M liters of milk daily, but every Friday, they refuse to give milk. Instead, they spend the day doing yoga, gossiping, and relaxing.

Given that **today is Saturday**, calculate how many liters of milk Thanos can collect from his cows after N days while respecting their no-milk-Friday rule.

Input
The first line contains an integer $T(1 \le T \le 1000)$: the number of test cases.

For each test case, the input consists of a single line containing three integers:$K(1 \le K \le 10000)$, $M(1 \le M \le 200)$, and $N(1 \le N \le 10^7)$:

- K: the number of cows Thanos owns.
- M: the liters of milk each cow produces per day.
- N: the number of days Thanos wants to collect milk.

Output
Output a single integer — the total liters of milk Thanos can collect from his cows after N days, respecting their "no-milk-Friday" rule.

input
2 5 10 10 3 20 15
output
450 780

D. Prime? Baby Prime?

3 s., 256 MB

Problem Statement

You are given an array of n positive integers: $a_1, a_2, \ldots, a_n$ where $1 \le a_i \le 1000$ for all $1 \le i \le n$.

Your task is to find the **maximum value of** $i + j$ such that the elements a_i and a_j are **coprime**, i.e., $\gcd(a_i, a_j) = 1$. If no such pair (i, j) exists, output -1.

Input
The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 10$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($2 \le n \le 2 \cdot 10^5$) — the length of the array.

The following line contains n space-separated positive integers a_1, a_2,..., a_n ($1 \le a_i \le 1000$) — the elements of the array.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output
For each test case, output a single integer — the maximum value of $i + j$ such that i and j satisfy the condition that a_i and a_j are coprime, or output -1 in case no i, j satisfy the condition.

input
6 3 3 2 1 7 1 3 5 2 4 7 7 5 1 2 3 4 5 3 2 2 4 6 5 4 3 15 12 16 5 1 2 2 3 6
output
6 12 9 -1 10 7

For the first test case, we can choose $i = j = 3$, with sum of indices equal to 6, since 1 and 1 are coprime.

For the second test case, we can choose $i = 7$ and $j = 5$, with sum of indices equal to $7 + 5 = 12$, since 7 and 4 are coprime.

E. Chada Baz

1 second, 256 megabytes

Author : Tiny Coder

Intake : 49

Dept : CSE

Mr. Tom and **Mr. Jerry** are two most wanted underworld don of **Dogland**. They constantly fight because they are rivals. Recently, They are fighting about money collection from the Dogland Bazar. One day they went to the city mayor **"Spike the bulldog"** for judgment. After considering their situation Spike issued them with two rules.

1. Tom can only collect money from the stores having **odd serial number**.
2. Jerry can only collect money from the stores having **even serial number**.

After some Days, Jerry wanted to know if Tom collect more money than him. Tom will receive more money than Jerry if the total amount of money collected from all stores with odd numbers is strictly greater than the total amount of money collected from all stores with even numbers. So, He investigates which store pays how much. He hires you to determine if Tom gets more than him.

You are given an array of integer number denoting amount each store pays. Index number of this array represents serial number of these stores. Print **"YES"** without quotation if Tom get more money then him. Other wise print **"NO"** without quotation.

Input

First Line of the Input contains a single integer $\mathbf{n}(1 \leq n \leq 1000)$, denoting Total number of stores in Dogland.

Second Line of the input contains $\mathbf{n}$ integer numbers $\mathbf{a_1, a_2, a_3, \ldots, a_n}$. Where $\mathbf{a_i}$ denotes how much the $\mathbf{i^{th}}$ store pays.$(0 \leq a_i \leq 1000)$

Output

Print **"YES"** without quotation if Tom get more money then him.

Other wise print **"NO"** without quotation.

input
7 101 656 453 98 453 234 456
output
YES

input
5 10 20 5 10 15
output
NO

input
6 10 10 10 10 10 10
output
NO

In the first Example, Tom's total collection is $101 + 453 + 453 + 456 = 1463$.

On the other hand, Jerry's total collection is $656 + 98 + 234 = 988$.

As Tom's collection is strictly greater than Jerry ($1463 > 988$), Output will be "YES".

F. FE!N 2

1 s., 256 MB

Author :Hamza Miraz (CSE BUBT,intake-48)

A grand concert is being held in the city named HAUKAU Land, attracting massive crowds of people. The crowd, overwhelmed by excitement, is loudly and crazily shouting "FE!N FE!N", causing the land to shake. The intensity of their screams is so powerful that any building located within d distance from the concert area collapses.

The concert located at (x, y). There are n buildings in the city. The $i'th$ building is located at (x_i, y_i).

Your task is to determine which buildings remain standing after the concert.

Input

The input consists of multiple test cases:

The first line contains an integer $t(1 \leq t \leq 100)$ — the number of test cases.

For each test case:

The first line contains four integers $n\ d\ x\ y$ $(1 \leq n \leq 10^5, 1 \leq d \leq 10^9, |x|, |y| \leq 10^9)$ — the number of buildings, the maximum distance their screams can be heard, and the location (x, y) of the concert.

Then follow n lines, each containing two integers $x_i\ y_i$ $(|x_i|, |y_i| \leq 10^9)$ — the location of the $i'th$ building.

It is guaranteed that the sum of all n across all test cases does not exceed 5000000.

Output
For each test case:

If all buildings collapse, then print "$FE!N$".

Otherwise, print:

1. An integer k, the number of buildings that remain standing after the concert.
2. A list of k integers: the $1-based$ indices of the remaining buildings.

input
2 4 5 0 0 1 1 4 4 6 8 0 3 3 10 5 5 10 10 0 0 5 15
output
2 2 3 FE!N

Test Case 1:

Building at location $(1, 1)$ collapses $(distance \approx 1.41 \leq 5)$. Buildings at locations $(4, 4)$, $(6, 8)$, and $(0, 3)$ have $distances \approx 5.66, 10, and$ 3 respectively. Buildings at indices 2 and 3 remain standing $(distances > 5)$.

Test Case 2:

All buildings are within the range of screams (distance from concert to each building is ≤ 10), so all collapse. Output is "$FE!N$".

G. Black Market Dealer

1 second, 256 megabytes

Bob is a black market dealer who buys products from an official store and resells them in the black market. The store has n products, each with a specific price. Bob has unlimited funds and must buy exactly k products from the store to resell in the black market.

Your task is to calculate the maximum profit Bob can earn after purchasing and reselling exactly k products.

Input
The input consists of three lines:

The first line contains two integers n and k, representing the number of products in the store and the number of products Bob must buy $(1 \leq k \leq n \leq 2 \cdot 10^5)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^9)$, where a_i represents the price of the i-th product in the store.

The third line contains n integers $b_1, b_2, \ldots, b_n$ $(1 \leq b_i \leq 10^9)$, where b_i represents the reselling price of the i-th product in the black market.

Output
Output a single integer — the maximum profit Bob can earn after reselling k products.

input
4 3 1 2 3 4 8 8 8 8
output
18

input
2 1 3 3 2 2
output
-1

H. Juice Harmony

1 second, 256 megabytes

Author :Hamza Miraz (CSE BUBT,intake-48)

There are n glasses of juice placed in a row. Each glass has some amount of juice in it.

Two friends — Alice and Bob — are at a juice party.

Alice only drinks from glasses at **odd positions (1st, 3rd, 5th, ...)**.

Bob only drinks from glasses at **even positions (2nd, 4th, 6th, ...)**.

They want to find a contiguous subarray of these glasses (one after another) such that both of them drink the same amount of juice from their allowed positions.

More formally, find out if there exists two indices l , r such that $1 \leq l \leq r \leq n$, and $a_l + a_{l+2} + a_{l+4} + \cdots + a_r = a_{l+1} + a_{l+3} + \cdots + a_{r-1}$ if l and r have the same parity and $a_l + a_{l+2} + a_{l+4} + \cdots + a_{r-1} = a_{l+1} + a_{l+3} + \cdots + a_r$ otherwise.

Input
The first line contains a single integer $t(1 \leq t \leq 10^4)$ — the number of test cases.

The first line of each test case contains a single integer $n(1 \leq n \leq 2 \cdot 10^5)$ — the total number of glasses.

The second line of each test case contains n integers $a_1, \ldots, a_n(1 \leq a_i \leq 10^9)$ — the amount of juice in each glass.

The sum of n over all test cases does not exceed $2 * 10^5$

Output
For each test case, output "YES" if there exists a subarray satisfying the condition, and "NO" otherwise.

input
6 3 1 3 2 6 1 1 1 1 1 1 10 1 6 9 8 55 3 14 2 7 2 8 1 2 11 4 1 5 1 2 6 2 6 1 5 7 8 9 2 5 10 4 4 9 6 7 8

output
YES YES NO YES NO YES

1st test case: a = [1, 3, 2]

Choose l = 1, r = 3 → Alice drinks $a_1 + a_3$ = 1 + 2 = 3, Bob drinks a_2 = 3 → Equal, output " YES ".

2nd test case: a = [1, 1, 1, 1, 1, 1]

Choose l = 2, r = 5 → Alice drinks $a_3 + a_5$ = 1 + 1 = 2, Bob drinks $a_2 + a_4$ = 1 + 1 = 2 → Equal, output " YES ".

3rd test case: a = [1, 6, 9, 8, 55, 3, 14, 2, 7, 2] **, no such contiguous subarray works**,output " NO "

4th test case: a = [1, 2, 11, 4, 1, 5, 1, 2]

Choose l = 2, r = 8 → Alice drinks $a_3 + a_5 + a_7$ = 11 + 1 + 1 = 13, Bob drinks $a_2 + a_4 + a_6 + a_8$ = 2 + 4 + 5 + 2 = 13 → Equal, output " YES ".

I. Binary Steps

1 second, 256 megabytes

You are given an array A of length n, where A[i] (1 ≤ i ≤ n) denotes the cost of landing on the i-th position. You start by landing on position 1, and your goal is to reach position n.

From any position, the player may jump ahead by a distance equal to a power of 2. That is, from position i, the player may move to positions i + 1, i + 2, i + 4, i + 8, etc., as long as the landing position does not exceed n.

Your task is to determine the minimum total cost needed to travel from position 1 to position n.

Input

The first line contains an integer n (1 ≤ n ≤$2*10^5$) — the number of positions.

The second line contains n space-separated integers A_1, A_2, ..., A_n (1 ≤ A_i ≤10^9) — the cost of landing on each position.

Output

Print a single integer — the minimum total cost to reach position n from position 1.

input
10 1 8 2 9 3 10 2 7 9 1
output
10

For TestCase 1 :

One possible optimal path:

Position 1 → 2 → 10

Total cost = a[1] + a[2] + a[10] = 1 + 8 + 1 = 10