

 **GDSC USICT PRESENTS**

<CODENITE/>

 **GAME OF CODE...**

**UPSOLVING
SESSION** 



SEPTEMBER 12, 2021



04:00 P.M.

```
using namespace std;
string
typename Traits>
information{
    te:
    traits pf_name;
    traits pf_family;

    ic:
    void set_information(Traits org_name, Traits org_family) {
        pf_name = org_name;
        pf_family = org_family;
    }

    void get_information() {
        std::cout << "Your name is " << pf_name << endl;
    }
}

int main(int argc, char* argv[]) {
    decltype(0);
    Information<std::string> a_person;

    a_person.set_information("Miled", "Kohsari");
    a_person.get_information();

    return 0;
}
```

we will start
at 4:05 PM



A. Angry Professor at USICT

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Once upon a time, There was an angry professor at **USICT** (Not GGSIPU).

He/She wanted to take as many classes as possible, but as you know the students wanted to bunk as many classes as possible.

Students were happy if they were able to bunk at least half of the classes in the entire semester but in that situation the teacher becomes angry.

There are **N** classes to be held in the entire semester **officially** : p but students tried their best to bunk as many classes as possible. :)

After the end of the semester, We need to figure out whether the students were happy or not.

Input

The first line contains an integer **N** ($1 \leq N \leq 10^5$) — Classes to be held in the entire semester (officially).

The second line contains **N** integers a_1, a_2, \dots, a_N : where $a_i = 0$ denotes that i^{th} class wasn't bunked and $a_i = 1$ denotes that i^{th} class was bunked.

Output

Output one string - **Yes** (if the students were happy) or **No** (if the students weren't happy).

Examples

input	Copy
2 0 1	
output	Copy
Yes	
input	Copy
5 0 1 0 1 0	
output	Copy
No	

Judge
Thi
(de
Thi
by
lar
the
sei
Virt
pas
par
ICP
see
not
arc
pro
is r
arc
rea
oth
You

$$C_0 \rightarrow X$$

$$C_1 \rightarrow \checkmark$$

$$C_1 \geq \frac{C_0 + 9}{2}$$

$$C_1 \geq C_0 + 9$$

$$C_1 \geq C_0$$

B. Unique Couples

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Our college, USICT has a problem when it comes to the sex ratio of classes, i.e the number of girls is usually way less than the number of boys (even 0 sometimes). So you are curious about how many unique boy-girl couples can be formed in your class.

Given the number of boys and girls in your class. Print the number of unique couples that can be formed.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 5000$) — the number of test cases. The description of the test cases follows.

Each line contains two numbers N ($0 \leq N \leq 10^9$) and M ($0 \leq M \leq 10^9$). Where N and M are the number of boys and number of girls in your class respectively.

Output

Print the total number of couples that can be formed.

Examples

input	Copy
2 1 4 3 7	
output	Copy
4 21	

input	Copy
3 1 5 0 2 3 0	
output	Copy
5 0 0	

#C

Judgm.

→ Ab

This cc
(deper
The sy
by the
langua
the tin
second

→ Vir

Virtual
past c
partici
ICPC m
seen ti
not for
archive
problem
is not
archive
read th
other ;

→ Ck

You ca

→ Im

You ca

$$\begin{aligned}
 & \text{int} \leftarrow 10^9 \quad 10^9 \\
 & \text{N boys} \quad M \text{ Girls} \\
 & 1 \text{ boy} \quad 1 \text{ girl} \\
 & N C_1 \times M C_1 = NM \\
 & \text{boy boy}
 \end{aligned}$$

$$\text{int} \times \text{int} \rightarrow (\text{int})$$

$$\begin{aligned}
 & \text{boy boy} \\
 & \text{boy girl} \\
 & \text{girl girl} \\
 & \text{int } n; \\
 & \text{int } n; \\
 & \text{cin } >> n >> n; \\
 & \text{cout } << n * n;
 \end{aligned}$$

D. Bar Pattern

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given an integer n (size of the array) and an array of integers a_1, a_2, \dots, a_n .

Print the required **Bar graph** where the **th** column contains exactly a_i stars(*) ($1 \leq i \leq n$).

Input

The first line contains a single integer n ($3 \leq n \leq 2 \cdot 10^5$), n — the size of the array.

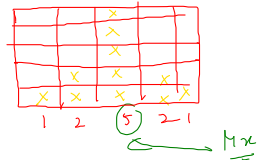
The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i \leq 49$) — where a_i — Height of the i -th column.

Output

For each test case print the required bar pattern as shown in the example.

Examples

Input	Output
5 3 5 5 2 1	<pre> * * * * * * *</pre>
5 1 2 5 2 1	<pre> * * * * * * *</pre>

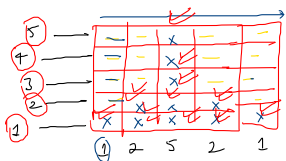


```

for i: 0... N-1 {
  for j: 0... Max {
    if (A[i] > Max)
      cout << 'x';
    else
      cout << ' ';
  }
  cout << '\n';
}

```

Row = Max
Col = N



```

for col: 0... N-1
  for j: 0... A[col] K = Max-1
    Pattern[k][col] = 'x';
    k--;

```

Array

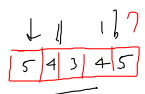
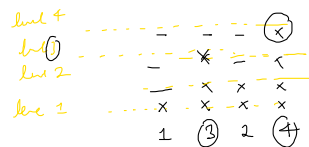
x 2D array

Max = 5

```

for knd: Max... 1 :
  for j: 0... N-1
    if (knd > A[j])
      cout << ' ';
    else
      cout << 'x';

```



F. Range Sum

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array of size n and q queries to perform on it.

Each query is as follows:

L, R, x : Take the subarray from L to R , a_L, a_{L+1}, \dots, a_R and add x to every element.

Print the resulting array after performing all the queries.

Input

The first line contains 2 integers n ($1 \leq n \leq 10^5$) and q ($1 \leq q \leq 10^5$).

The second line contains n integers a_1, a_2, \dots, a_n ($-10^9 \leq a_i \leq 10^9$) followed by q lines.

Each line contains 3 integers L, R ($1 \leq L \leq R \leq n$) and x ($-10^9 \leq x \leq 10^9$).

Output

Print the array after updating all the queries.

Example

Input

```
10 4
1 2 3 4 5 6 7 8 9 10
2 5 1
3 7 1
9 9 -1
8 10 3
```

Output

```
1 7 9 10 11 7 8 11 11 13
```

Note

Initially: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

1st query: [1, 7, 8, 9, 10, 6, 7, 8, 9, 10]

2nd query: [1, 7, 9, 10, 11, 7, 8, 9, 10]

3rd query: [1, 7, 9, 10, 11, 7, 8, 8, 10]

4th query: [1, 7, 9, 10, 11, 7, 8, 11, 11, 13]

Diff. Array

system

L R x

for i: L -- R

$A[i] += x$

$O(Q \times N)$

$10^5 \times 10^5$

10^{10}

10m

+10

1 2 3 4 5

Prefix Sum

6 1 2 - - -
1 2 1 1 1

A

10

Prefix 1 3 6

$Prefix[i] = A[0] + A[1] + \dots + A[i]$

$O(Q \times 1 + N)$

$O(1)$

$O(Q \times R)$

$O(Q + N)$

if $(r+1 \leq n)$

$B[A] += x$

$B[k+1] -= x$

2×10^5

$10^5 + 10^6$

Jaideep



Kushagra



Parv



Google Developer Student Clubs
University School Of Information
Communication and Technology