Problem A. Knapsack Problem

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Today HH becomes a shopper, and he wants to buy a lot.

HH has a bag that can carry at most w kilograms things in total, and he has d dollars.

Now he wants to buy n items , the i^{th} item weights w_i kilogram and costs c_i dollars.

HH is not good at math so he asks you to tell him whether he can buy all the things and carry them with the bag.

Input

The first line contains an positive integer $T(1 \leq T \leq 10)$, represents there are T test cases.
For each test case:

The first line contains three positive integers $n, w, d(1 \leq n \leq 100, 1 \leq w \leq 100, 1 \leq d \leq 100)$ - the number of items HH wants to buy, the max weight that his bag can carry, and the money he has.

The second line contains n integers $w_1, w_2 \ldots w_n (1 \leq w_i \leq 100)$.

The third line contains n integers $c_1, c_2 \ldots c_n (1 \leq c_i \leq 100)$.

Output

For each test case, output one line "YES" (without quotes) if HH is possible to buy all the items and carry them in his bag, and "NO" (without quotes) otherwise.

Examples

Standard input	Standard input

2
4 12 17
1 2 4 5
5 4 6 2
4 11 17
1 2 4 5
5 4 6 2
	YES
	NO

Hint

In the first example all the items cost 17 dollars in total and weight 12 kilograms in total, HH has enough money and his bag can carry 12 kilogram things.

Problem B. Matrix

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Today HH is palying with a $n \times n$ matrix.

All the numbers of the matrix is 0 initial, and every time HH will do one of the following things:

1.make all the numbers in the k row become v
2.make all the numbers in the k column become v

Now HH wants to know what's the final matrix after q options.

Input

The first line contains an positive integer $T (1 \leq T \leq 10)$, represents there are T test cases.

For each test case:

The first line contains two positive integers $n, q (1 \leq n \leq 500, 1 \leq q \leq 2 \times 10^5)$ - the size of the matrix and the times of the options.

Then q lines following, for each line contains three integers $op, k, v (1 \leq op \leq 2, 1 \leq k \leq n, 1 \leq v \leq 100)$.

if $op = 1$, then HH will change all the numbers in the k row into v

if $op = 2$, then HH will change all the numbers in the k column into v

Output

For each test case, you should output n lines , each line n numbers , indicating the final matrix, note that for each line ,you should print exactly one blank between two numbers.

Examples

Standard input	Standard input

1
3 3
1 2 3
2 2 1
1 1 3
	3 3 3
	3 1 3
	0 1 0

Hint

In the first example

$$\begin{matrix}0&0&0\\0&0&0\\0&0&0\end{matrix} \rightarrow \begin{matrix}0&0&0\\3&3&3\\0&0&0\end{matrix} \rightarrow \begin{matrix}0&1&0\\3&1&3\\0&1&0\end{matrix} \rightarrow \begin{matrix}3&3&3\\3&1&3\\0&1&0\end{matrix}$$

scanf is commended

Problem C. Paint Box

Input file: standard input
Output file: standard output
Time limit: 5 second
Memory limit: 128 megabytes

We have n empty boxes, so let's recolor those boxeswith m colors.

The boxes are put in a line. It is not allowed to color any adjacent boxes with the same color. Boxes i and $i + 1$ are said to be adjacent for every $i, 1 \leq i \leq n$.

And we also want the total number of different colors of the n boxes being exactly k.

Two ways are considered different if and only if there is at least one box being colored with different colors.

Input

The first line of the input contains integer $T(1 \leq T \leq 100)$ -the number of the test cases
For each case:
there will be one line, which contains three integers
$n, m, k(1 \leq n, m \leq 10^9, 1 \leq k \leq 10^6, k \leq n, m)$.

Output

For each test case, you need print an integer means the number of ways of different coloring methods modulo $10^9 + 7$.

Examples

Standard input	Standard input

2
3 2 2
3 2 1
	2
	0

Hint

In the first example we have two ways:

121

212

where 1 and 2 are two different color.

In the second example we can’t do that.

Problem D. Music Problem

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Listening to the music is relax, but for obsessive(强迫症), it may be unbearable.

HH is an obsessive, he only start to listen to music at $12:00:00$, and he will never stop unless the song he is listening ends at integral points (both minute and second are 0), that is, he can stop listen at $13:00:00$ or $14:00:00$,but he can't stop at $13:01:03$ or $13:01:00$, since $13:01:03$ and $13:01:00$ are not an integer hour time.

Now give you the length of some songs, tell HH whether it's possible to choose some songs so he can stop listen at an integral point, or tell him it's impossible.

Every song can be chosen at most once.

Input

The first line contains an positive integer $T(1 \leq T \leq 60)$, represents there are T test cases.

For each test case:

The first line contains an integer $n(1 \leq n \leq 10^5)$, indicating there are n songs.

The second line contains n integers $a_1, a_2 \ldots a_n$ $(1 \leq a_i \leq 10^9)$, the i^{th} integer a_i indicates the i^{th} song lasts a_i seconds.

Output

For each test case, output one line "YES" (without quotes) if HH is possible to stop listen at an integral point, and "NO" (without quotes) otherwise.

Examples

Standard input	Standard input
3	NO
3	YES
2000 1000 3000	YES
3	
2000 3000 1600	
2	
5400 1800	

Hint

In the first example it's impossible to stop at an integral point.

In the second example if we choose the first and the third songs, they cost 3600 seconds in total, so HH can stop at $13:00:00$

In the third example if we choose the first and the second songs, they cost 7200 seconds in total, so HH can stop at $14:00:00$

Problem E. Shortest Path

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Today HH becomes a designer, and he faces a problem so he asks you for help.

$Treeisland$ is a country with n cities and $n-1$ two-way road and from any city you can go to any other cities.

HH the designer is going to design a plan to divide n city into $n/2$ pairs so that the sum of the length between the $n/2$ pairs city is minimum.

Now HH has finished it but he doesn't know whether it's true so he ask you to calculate it together.

It's guaranteed that n is even.

Input

The first line contains an positive integer $T(1 \le T \le 100)$, represents there are T test cases.

For each test case:

The first line contains an positive integer $n(1 \le n \le 10^4)$, represents the number of cities in $Treeisland$, it's guarantee that n is even.

Then $n-1$ lines followed.

Each line contains three positive integer u, v and len, $(u \ne v, 1 \le u \le n, 1 \le v \le n, 1 \le len \le 10^9)$indicating there is a road of length len between u and v.

It's guarantee you can get to any city from any city.

Output

For each test case, output in one line an integer, represent the minimum sum of length.

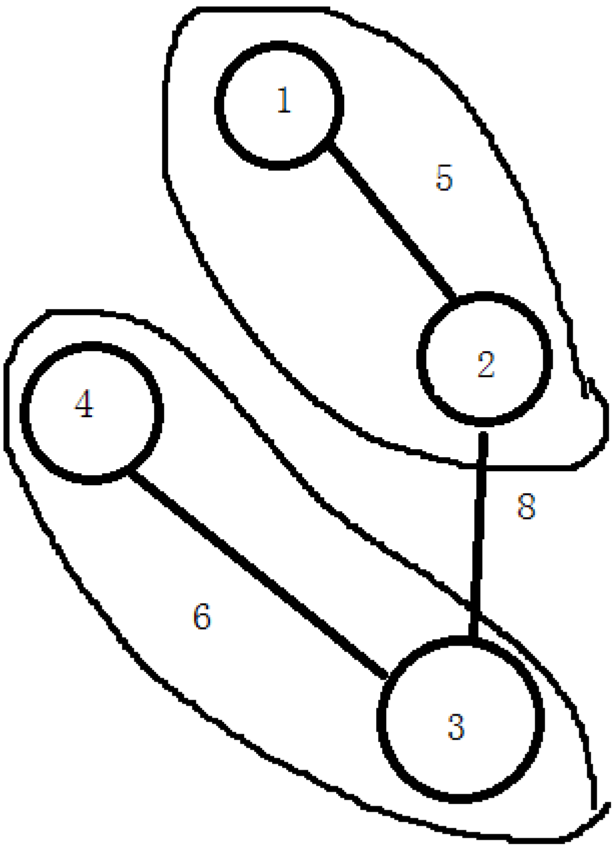
Examples

Standard input	Standard input

2
4
1 2 5
2 3 8
3 4 6
6
1 3 5
3 2 3
4 5 4
4 3 9
4 6 10
	11
	31

Hint

In the first example, you can divide them into $(1, 2)$, and $(3, 4)$, then the minimum sum of length is $5 + 6 = 11$



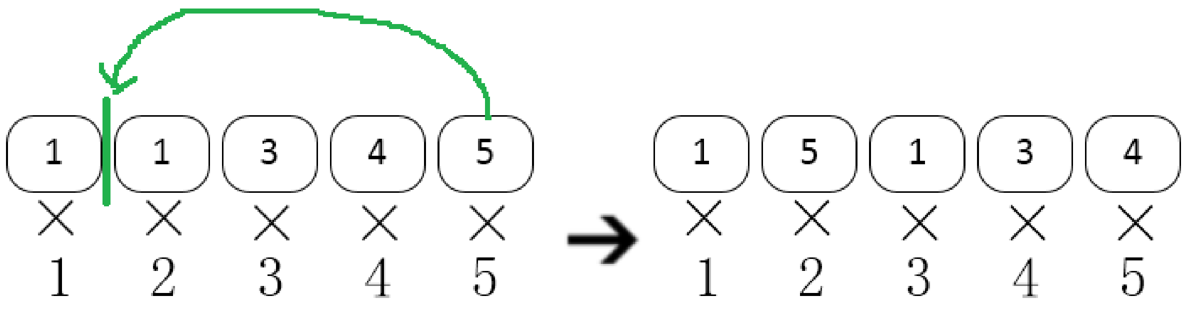
In the second example, you can divide them into $(1, 3)$,$(2, 4)$,$(5, 6)$, hen the minimum sum of length is $(5 + 3) + (9) + (10 + 4) = 31$

Problem F. Maximize The Beautiful Value

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Today HH finds a non-decreasing sequence$(a_1, a_2 .... a_n, a_i \le a_{i+1})$, he thinks it's not beautiful so he wants to make it beautiful.

To make it, HH will choose exactly one number and move it forward at least k steps(i.e. you can move a_i to a_j if $k \le i - j$), and then he defines the beautiful value $F(n)$ as $\sum_{i=1}^{n} i \times a_i$.



HH asks you to calculate $max(F(n))$

Input

The first line contains an positive integer $T(1 \le T \le 10)$, represents there are T test cases.

For each test case:

The first line contains two positive integers $n, k(1 \le n \le 10^5, 1 \le k < n)$, the length of the sequence ,the least steps you need to move.

The second line contains n integers $a_1, a_2 \ldots a_n (1 \le a_i \le 10^8)$ - the sequence.

Output

For each test case, you should output the max $F(n)$.

Examples

Standard input	Standard input

3
5 3
1 1 3 4 5
5 2
1 1 3 4 5
5 1
1 1 3 4 5
	46
	50
	53

Hint

In the first example, you can move the fifth number 4 for 3 steps and make the sequence become $[4, 1, 1, 3, 5]$, then the beautiful value is $4 \times 1 + 1 \times 2 + 1 \times 3 + 3 \times 4 + 5 \times 5 = 46$.

You can also move the fifth number to make it become $[1, 5, 1, 3, 4]$, the beautiful value is also 46.

In the second example, you can move the fifth number 5 for 2 steps and make the sequence become $[1, 1, 5, 3, 4]$

In the second example, you can move the second number 1 for 1 steps and then the sequence is still $[1, 1, 3, 4, 5]$

scanf is commended。

Problem G. Maximize The Beautiful Value++

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

The term of this problem is the same as the problem F, the only exception — The sequence may not necessary to to non-decreasing.

Input

The first line contains an positive integer $T(1 \leq T \leq 10)$, represents there are T test cases.

For each test case:

The first line contains two positive integers $n, k(1 \leq n \leq 10^5, 1 \leq k < n)$, the length of the sequence ,the least steps you need to move.

The second line contains n integers $a_1, a_2 \ldots a_n(1 \leq a_i \leq 10^8)$ - the sequence.

Output

For each test case, you should output the max $F(n)$.

Examples

Standard input	Standard input

4
5 3
1 1 3 4 5
5 2
1 1 3 4 5
5 1
1 1 3 4 5
5 3
3 5 4 1 1
	46
	50
	53
	43

Hint

In the forth example, you can move the fifth number 1 for 4 steps and then the sequence becomes $[1, 3, 5, 4, 1]$

Problem H. Magic Maze

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

There is a magic maze that its roads is unidirectional and you will not arrive the same resting area if you walk along the road (the maze is acyclic). There are n resting areas and m roads in the maze. Some roads make you get treasure, while others make you lost treasure. You should pick the place to set out and get treasure as much as possible.

Note that for each road you can go through only once.

Input

The first line: the number of case T ($1 \le T \le 110$)
In each test case:
The first line is two integers: the number of resting area n, the number of roads m($1 \le n \le 1000$, $0 \le m \le n \times (n-1) \div 2$)
m lines follow, each with three integers: the beginning u, the end v, treasure w($0 \le u < n, 0 \le v < n, -1000 \le w \le 1000$)

Output

T lines, each with an integer what is the maximum treasure.

Examples

Standard input	Standard input

2
5 4
0 1 -10
1 2 10
2 3 10
3 4 -10
4 4
0 1 4
0 2 5
2 3 -2
3 1 4
	20
	7

Hint

In the first example, you can go 1 -> 2 -> 3, the ans is $10 + 10 = 20$

In the second example, you can go 0 -> 2 -> 3->1, the ans is $5 - 2 + 4 = 7$

Problem I. HGCD

Input file: standard input
Output file: standard output
Time limit: 10 second
Memory limit: 128 megabytes

HH is learning GCD today, GCD is short for $Greatest\ Common\ Divisor$. You can look at this implementation:

```
long long gcd(long long a, long long b){
    while(a > 0 && b > 0){
        a %= b;
        swap(a , b);
    }
    return a + b;
}
```

But HH is so careless that he changes “%” into “-“, let’s call this new funtion $HH's\ Greatest\ Common\ Divisor\ (HGCD)$. You can look at this implementation:

```
long long hgcd(long long a, long long b) {
    while (a > 0 && b > 0) {
        a -= b;
        swap(a , b);
    }
    return a + b;
}
```

Now HH run $HGCD(a, b)$ for all $1 \leq a \leq n$, $1 \leq b \leq n$,HH wants to know how many times does his algorithm - $HGCD$ works correctly.

Note:

```
void swap(long long &a, long long &b){
    long long t = a;
    a = b;b = t;
}
```

Input

The first line of input will contain an integer $T(1 \le T \le 5)$,denoting the number of test cases.
For each test case:
The first and only line contains an integer n ($1 \le n \le 10^{12}$)

Output

For every test case output the number of times $gcd(a,b) == hgcd(a,b)$
It’s guaranteed that long long is enough for this problem.

Examples

Standard input	Standard input
2 3 2	8 4

Hint

In the first example, we have $gcd(a,b) == hgcd(a,b)$ for all $1 \le a \le 3, 1 \le b \le 3$ except $a = 2, b = 3)$

Problem J. The Trip On Abandoned Railway

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

There are many ghosts at the abandoned station on unknown railway.
We mark the abandoned stations as $1, 2..n$ according to the order. There are a_i ghosts at the i^{th} station.$Yakumo\ Yukari$ often opens a black hole and makes a train appearing at a certain station. For example, the train appears at the x station, and k ghosts get off at the station. Then there would be $k+d$ ghosts at the $x+1$ station to get off,$k+2\times d$ at $x+2$ station and so on….There would be $k+y*d$ ghosts at the $x+y$ station to get off ($0 \le y, x+y \le n$). In others words, the numbers getting off at $x, x+1, x+2..n$ station form a tolerance of d arithmetic progression.(you can consider ghosts getting off at the same time.)
$Onozuka\ Komachi$ would comes a certain station to take away the ghosts.(The number of ghosts at the station would become 0)You have the records of trains appearing and $Komachi$ coming. You should tell $Komachi$ how much ghosts at a certain station when she come to there.

Input

The first line contains an positive integer $T(1 \le T \le 10)$, represents there are T test cases.
For each test case:
The first line contains three positive integers
$n, m, d(1 \le n \le 10^5, 1 \le m \le 10^5, 1 \le d \le 1000)$ - the number of station,the number of records,and the tolerance of the arithmetic progress.
The second line contains n integers $a_1, a_2 \ldots a_n(1 \le a_i \le 1000)$.
Then m lines followed.
Each line contains a records and there are two types.
1 x y,indicating train appearing at x station and y ghosts geting off.
2 x y,indicating Komachi coming to the x station.
$(1 \le x \le n, 0 \le y \le 1000)$

Output

For each second records($2 \quad x$), output an integer in one line, representing the number of ghosts at the station.Since the ans may be too large, out put tme ans mod 10^9+7.

Examples

Standard input	Standard input

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

Hint

There lists the number of ghosts changing at these station.

case1 :

```
1 2 3 3 2 1
2 4 6 7 7 7
0 4 6 7 7 7
0 0 6 7 7 7
0 0 0 7 7 7
0 0 0 0 7 7
0 0 0 0 0 7
```

case2 :

```
1 2 3 4 5
1 2 3 6 9
1 2 3 0 9
1 2 3 0 9
```

Problem K. Segment Tree

Input file: standard input
Output file: standard output
Time limit: 2 second
Memory limit: 128 megabytes

Today HH is learning a new data structure named $segment\ tree$, which is often used to solve segment problem, here comes one:

Gave you an unordered sequence of length n, $(a_1\ , a_2, \ldots, a_n)$, now you are supposed to calculate how many segment $[L, R](1 \le L \le R)$ are there satisfies two conditions :

the length of the segment is k(i.e. $R - L + 1 = k$).
the number between L and R(both including) appears at least q times in total.
HH thinks the problem is too easy so he gives the problem to you.

Input

The first line contains an positive integer $T(1 \le T \le 10)$, represents there are T test cases.

For each test case:

The first line contains three positive integers $n, k, q(1 \le n \le 100, 1 \le k \le 100, 1 \le q \le 100)$, the length of the sequence , the length of the segment $[l, R]$, and the times required to appear.

The second line contains n integers $a_1, a_2 \ldots a_n(1 \le a_i \le 100)$ - the sequence.

Output

For each test case, output one line an integer : the number of segment satisfies both conditions.

Examples

Standard input	Standard input

3
5 3 2
2 3 2 4 5
5 3 3
2 3 2 4 5
5 3 6
2 3 2 4 5
	4
	3
	0

Hint

In the first example , we can find 4 segments:

$[1,3]$,1 appears 0 time,2 appears 2 times,3 appears 1 time,so 3 times in total.

$[2,4]$,4 times in total.$[3,5]$ 3 times in total.$[4,6]$,2 times in total.

In the second example, we can find:

$[1,3],[2,4],[3,5]$.

In the third example, we can’t find any.