Problem A. Problem Express

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`White`

Islam and Zyad are participating in a programming contest as a team to win the time travel machine. There are a total of X problems in the contest.

Before the blind hour began:

— Zyad had solved a problems

— Islam had solved b problems

Determine how many problems remain for Zyad and Islam to solve together.

Input

The input consists of three integers: X $(10 \leq X \leq 20)$ — the total number of problems in the contest. a, b $(0 \leq a, b \leq X,\ a + b \leq X)$ — where a is the number of problems Zyad had solved, and b is the number of problems Islam had solved.

Output

Output how many problems remain for Zyad and Islam to solve together.

Example

standard input	standard output
15 6 3	6

Problem B. Oops!... They Did It Again

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Light Blue`

Long story short, Agamay and Moamen broke the time machine — again.

Turns out, they had severely underestimated the energy cost of stabilizing the timeline. Panicked, they called in Islam, the most pragmatic engineer they knew. Islam decided to approach the problem differently: he would reconstruct the entire timeline as a graph, with n historical checkpoints as nodes and m unstable connections between them.

Each edge connects two time points u_i and v_i, and stabilizing that link costs $u_i + v_i$ units of temporal energy. To optimize the machine's routing system, Islam needs the graph to be fully connected (so that every time period is reachable), contain no duplicate or invalid jumps (i.e., no multiple edges or self-loops), and above all — have the minimum possible energy cost.

Luckily, the machine allows him to tweak the edges: he can replace one end of any edge with any other node (as long as it stays valid), and he can do this operation as many times as needed.

Your task is to help Islam minimize the total energy cost to make the graph connected, or determine if it's impossible.

Formally, You're given a graph consisting of n nodes numbered from 1 to n, and m edges. The i-th edge connects the nodes u_i and v_i

(It's guaranteed that the graph doesn't have multiple edges or self-loops)

We define the cost of a graph to be the sum of the values on the ends of each edge.

More formally:

$cost = \sum_{i=1}^{m} u_i + v_i$

You are allowed to do the following operation any number of times (possibly zero).

Choose one index i $(1 \le i \le m)$ and change one of the ends of the $i-th$ edge.

For example, if the third edge was "1 3" then we can make it "1 x" or "x 3" $(1 \le x \le n)$.

After preforming the operations, the graph must be a connected graph, and its cost must be minimum, and it mustn't contain multiple edges or self-loops.

If the graph cannot be connected print -1. Otherwise, print the minimum possible cost of the resulting graph.

Input

Each test consists of multiple test cases. The first line contains a single integer t $(1 \le t \le 10^4)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and m $(2 \le n \le 2 \times 10^5)$ $\left(1 \le m \le \min\left(\frac{n(n-1)}{2}, 2 \times 10^5\right)\right)$ — the number of nodes, and edges in the graph respectively.

Each of the next m lines contains two integers u_i, v_i $(1 \le u_i, v_i \le n)$ — representing the i-th edge.

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

Output

If the graph cannot be connected print -1. Otherwise, print the minimum possible cost of the resulting graph.

Examples

standard input	standard output
1 5 6 1 2 1 3 1 4 1 5 2 5 3 5	29
1 7 6 1 2 1 3 2 4 2 5 3 6 3 7	33

Problem C. Viking Signals

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Purple`

Since Agamy likes Vikings, Moamen wanted to surprise him. So he set course for Scandinavia in the 9th century to explore the Viking Age.

While exploring a coastal village, they encountered a curious Viking custom: stone-signal communication. To send alerts across fjords (long, narrow sea inlets), the Vikings would drop enchanted stones into a calm pond. Each stone created a ripple that expanded at a speed of v meters per second. To maintain rhythm and avoid confusion, they dropped one stone every T seconds.

A stone dropped into the still pond creates a series of waves. The waves move outward at a speed of v meters per second. A new wave forms every T seconds. The farthest wave has currently reached a distance of R meters.

Find the number of fully formed waves that have been created.

Input

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

Each of the next t lines contains three integers v, T, and R ($1 \le v, T, R \le 10^9$) — the speed of wave propagation in meters per second, the time interval between the creation of waves in seconds, and the current radius of the farthest wave in meters.

Output

For each test case, print a single integer — the number of fully formed waves.

Examples

standard input	standard output
1 2 2 20	5
3 2 4 320 3 243 1458 84 8 245446656	40 2 365248

Problem D. Temporal Paradox

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Gold`

Agamy and Moamen couldn't agree on their next time-travel destination. So, as all reasonable time travelers do, they decided to settle it with a game.

They found a collection of stone piles in a deserted marketplace and agreed on the following rules:

There are N piles of stones. The i-th pile initially contains a_i stones.

Two players take turns alternately. On each turn, a player selects any pile and removes any number of stones strictly less than the value of the most significant bit (MSB) of the current number of stones in that pile.

The MSB is defined as the largest power of two that does not exceed the current number. For example:

$\text{MSB}(6) = 4$

Players alternate turns, and the player who cannot make a move loses.

Assuming both players play optimally, determine who will win — the first player (who starts the game) or the second.

Input

The first line contains a single integer t ($1 \le t \le 100$) — the number of test cases.

The first line of each test case contains a single integer N ($1 \le N \le 10^5$) — the number of piles.

The second line contains N integers $a_1, a_2, \ldots, a_N$ ($1 \le a_i \le 10^9$) — the initial number of stones in each pile.

It is guaranteed that the sum of N over all test cases does not exceed 10^5.

Output

For each test case, print **First if the first player wins, or Second if the second player wins.**

Example

standard input	standard output
2 1 5 2 6 14	First Second

Problem E. The Infected Tree

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Orange`

Zyad wanted a break — a place with fresh air, serene hills, and the kinds of trees he loved so much. He set the time machine to ancient Mesoamerica, hoping to breathe under the canopies of nature.

What he didn't expect to find was a rare tree species among the Aztecs — a tree of immense historical and biological value. But something was wrong: the tree was suffering from a mysterious infection, with some of its branches (nodes) clearly affected.

Zyad wants to study the spread of the infection. The tree has n nodes and m of them are known to be infected. For analysis, he defines the **infection density** for any array x of nodes as:

$$f(X) = \frac{\text{number of infected nodes in x}}{\text{number of nodes in x}}$$

You are given q queries. Each query gives two nodes u and v on the tree. Consider the unique simple path between them. Zyad wants to know the sum of $f(x)$ over all subsegments of that path modulo $10^9 + 7$.

Input

- The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases.
- For each test case:
  - The first line contains three integers n, m, and q ($1 \le n \le 10^5$, $0 \le m \le n$, $1 \le q \le 10^5$) — the number of nodes in the tree, the number of infected nodes, and the number of queries.
  - The second line contains m distinct integers ($1 \le b_i \le n$) — representing the numbers of the infected nodes. If $m = 0$, this line is empty.
  - The next $n - 1$ lines each contain two integers u and v ($1 \le u, v \le n$) — representing an undirected edge between u and v.
  - The next q lines each contain two integers u and v ($1 \le u, v \le n$) — specifying the query nodes.
- It is guaranteed that the sum of n and q does not exceed 2×10^5.
- It is guaranteed that the sum of the number of infected nodes in all test cases across all query paths does not exceed 2×10^5.

Output

For each query, output a single integer — the sum of $f(x)$ over all subsegments of the simple path from u to v **modulo** $10^9 + 7$.

Example

standard input	standard output
2 11 5 5 2 5 7 8 11 1 2 1 3 2 4 3 5 3 6 5 7 5 8 6 9 6 10 10 11 2 5 7 11 3 5 7 8 9 10 5 5 2 1 2 3 4 5 1 2 2 3 3 4 4 5 1 5 2 4	166666672 433333345 500000005 6 0 15 6

Problem F. Saving Energy

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Black`

After debugging their machine, Agamay and Moamen were finally ready to start their journey through history. But before randomly jumping through wormholes, they decided to carefully plan their path.

They charted out all the eras they wanted to visit and represented each as a vertex in a complete graph with N vertices. Every vertex corresponds to a specific time period.

To conserve energy, they decided to only use paths formed within a **spanning tree** of the graph. A spanning tree ensures that every era is connected with exactly $N - 1$ time jumps and no loops — perfect for keeping the timeline stable.

In each spanning tree, there's a unique path from the first vertex (where their journey begins) to the N-th vertex (where it ends). The time machine consumes exactly 1 unit of energy for each edge crossed on this path.

Since there's a huge number of possible spanning trees, and each is equally likely to occur due to quantum fluctuations, they want to compute the **expected energy cost** of traveling from vertex 1 to vertex N.

Can you help them calculate the expected number of edges on the journey from the first to the last time period, averaged over all possible spanning trees?

Formally, consider a complete graph on N vertices. Pick a spanning tree uniformly at random. What is the expected length of the unique simple path from vertex 1 to vertex N?

Since the answer can be a rational number $\frac{P}{Q}$ (with Q coprime to 998244353), output the value of $P \cdot Q^{-1} \bmod 998244353$, where Q^{-1} is the modular inverse of Q modulo 998244353.

Input

The first line contains a single integer T ($1 \le T \le 10^5$), the number of test cases.

Each of the next T lines contains a single integer N ($2 \le N \le 5 \cdot 10^5$) — the number of vertices in the complete graph.

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

Output

For each test case, output a single integer — the expected length of the path from vertex 1 to vertex N, taken modulo 998244353.

Example

standard input	standard output
3 2 3 7	1 332748119 658091716

Problem G. Mansa Musa

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Green`

Mazen, ever fascinated by the mysteries of wealth, sought out the land once considered the richest in history — Mali. Intrigued by the empire's immense gold reserves and curious about how such value could be stored, he traveled back in time and met Mansa Musa himself. Upon arriving, he was asked to help the empire solve a peculiar storage optimization problem, one involving arranging precious items in a very specific order to maximize total value.

You are given a Sorted Permutation p of length n, and a matrix a of n rows and n columns.

Let $a_{i,j}$ denote the element of the intersection of the *i-th* row with the *j-th* column in a.

A permutation of length n is an array consisting of n distinct integers from 1 to n in any order. For example, $[2, 3, 1, 5, 4]$ is a permutation, but $[1, 2, 2]$ is not a permutation (as 2 appears twice in the array) and $[0, 1, 2]$ is also not a permutation (as $n = 3$, but 3 is not present in the array).

To solve the problem, Mazen walks on p from left to right and assigns each element either the color black or white.

Let p' be an empty array; after the coloring, the following happens:

- Let $b_1, b_2, \ldots, b_x$ denote the white-colored elements in increasing order, $0 \leq x$;
- Let $c_1, c_2, \ldots, c_y$ denote the black-colored elements in increasing order, $0 \leq y$;
- Then $p' = [b_1, b_2, \ldots, b_x, c_1, c_2, \ldots, c_y]$, where $x + y = n$.

Calculate the maximum possible sum of $a_{p'_i,i}$ Mazen can achieve after performing the described operation.

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 300$) — the rows and columns of the matrix a.

Each of the following n lines contains n integers $a_{i,1}, a_{i,2}, \ldots, a_{i,n}$ ($-10^5 \leq a_{i,j} \leq 10^5$) — the elements of the matrix a, where i is the current row.

It is guaranteed that the sum of n over all test cases does not exceed 300.

Output

For each test case, print the answer on a single line.

Example

standard input	standard output
3 2 1 2 3 4 2 1 4 3 2 3 1 2 3 4 5 9 1 7 9	5 7 17

Problem H. Lost Library of Córdoba

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

Zyad once heard tales of the legendary **Lost Library of Córdoba**, said to hold secrets from a golden age. Driven by curiosity, he scribbled down the coordinates and the date, then set the time machine in motion toward ancient Al-Andalus.

Deep inside the ruins, Zyad discovered an old paper. Upon closer inspection, he realized it sayed something... a hidden message in the form of strings!

You are given an integer N and an $N \times N$ matrix a. You must count the number of strings s of length N over the lowercase English alphabet (a–z) such that for all $1 \le i, j \le N$, $a_{i,j} = \mathrm{LCP}\big(\mathrm{suff}_i(s), \mathrm{suff}_j(s)\big)$, where:

- $\mathrm{suff}_i(s)$ (the i-th suffix of s) is the substring $s_i s_{i+1} \dots s_N$.
- $\mathrm{LCP}(x, y)$ (the longest common prefix of strings x and y) is the largest integer ℓ such that the first ℓ characters of x and y coincide.

Output the number of valid strings modulo $10^9 + 7$.

Input

The first line contains a single integer T ($1 \le T \le 500$), the number of test cases. Each test case begins with an integer N ($1 \le N \le 500$), the length of the string and the dimension of the matrix.

Then follow N lines, each containing N integers $a_{i,1}, a_{i,2}, \dots, a_{i,N}$ ($0 \le a_{i,j} \le N$), describing the LCP matrix.

It is guaranteed that there is at least one string that satisfies the given LCP array.

It is guaranteed that the sum of all N over all test cases does not exceed 500.

Output

For each test case, output one integer: the number of strings s of length N over the lowercase English alphabet such that for all $1 \le i, j \le N$, $a_{i,j} = \mathrm{LCP}\big(\mathrm{suff}_i(s), \mathrm{suff}_j(s)\big)$, taken modulo $10^9 + 7$.

Example

standard input	standard output
2 2 2 0 0 1 2 2 1 1 1	650 26

Problem I. All Roads Lead to Rome

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Blue`

While watching a dull football match, Mazen got bored and decided he wanted to see some *real* sports. Since the time machine was now available to everyone, Agamy couldn't stop him.

He set the destination: Ancient Rome.

There, in the roaring Colosseum, Mazen witnessed a brutal but exciting gladiatorial contest. Thrilled beyond belief, he wanted to capture that moment forever. So he pulled out his camera from the future and asked everyone in the arena to pose.

The camera takes a circular photo, initially centered at $(0, 0)$ with radius L. To make sure it includes everyone, Mazen can:

- Double the current radius: $L \leftarrow L \times 2$
- Add k to the radius: $L \leftarrow L + k$

You're given the coordinates of n people. Find the minimum number of operations Mazen must perform so that all people are within the camera's range.

Input

Each test contains multiple test cases. The first line contains the number of test cases T ($1 \le T \le 10^6$). The description of the test cases follows. The first line contains three integers n, L, k ($1 \le n \le 10^6$; $1 \le L, k \le 10^9$).

Each of the next n lines contains two integers x_i and y_i ($|x_i|, |y_i| \le 10^9$).

It is guaranteed that the sum n across all test cases does not exceed 10^6.

Output

For each test case, Print one integer: the minimum number of operations needed.

Example

standard input	standard output
4 1 10 5 2 2 1 1 1 3 4 2 1 1 0 0 0 10 1 1 1 1000000000 1000000000	0 3 4 31

Problem J. Trading Routes

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

Zyad was so fascinated by the great trading routes of Ancient Egypt that he decided to travel there himself using the time machine to examine them up close.

While analyzing trade efficiency, he stumbled upon a mathematical challenge related to dividing a route for optimal value.

You're given a permutation of size n and you have to choose l, r $(l < r < n)$ to minimize the score of the permutation where the score is :

- $mx(1, l) + mx(l + 1, r) + mx(r + 1, n)$

where $mx(a, b)$ is the maximum element in the range $[a, b]$.

Input

The first line contains one integer n $(3 \le n \le 2 * 10^5)$. The second line contains n distinct positive integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$, the elements of the permutation.

Output

Output the minimum score you can obtain.

Example

standard input	standard output
6 5 2 1 6 3 4	12

Problem K. Good news, everyone!

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Cyan`

Agamay and Moamen are two dudes who just *love* watching `Futurama`. During one of their late-night rewatches, they got struck by a genius idea—why not build their own time machine like the one Professor Farnsworth uses?

After days of tinkering, soldering, and coding, the machine was finally ready. They flicked the switch. A buzz, a spark, and then... chaos.

While Moamen was busy aggressively kicking the side of the malfunctioning time machine, Agamay decided to debug the core code. It turns out the problem was buried somewhere in the **time array**, a mysterious data structure holding quantum timestamps. To figure out what went wrong, Agamay needs to compute something called the **Longest Common Prefix (LCP)** of different parts of the array under random quantum fluctuations.

Can you help him?

You are given an array a of length n. We define the Longest Common Prefix (LCP) of two arrays x and y as the length of the longest sequence starting from the first element where x and y are identical. Formally, the LCP of x and y is the largest integer k (possibly zero) such that $x_i = y_i$ for all $1 \le i \le k$.

You need to process q queries. In each query, you are given two non-intersecting ranges of the array. After shuffling the entire array uniformly at random, find the **expected value** of the LCP of the two given ranges.

Since the answer can be a fraction, you need to output the result modulo 998244353. More precisely, if the expected value is a rational number $\frac{p}{q}$, you should output $p \cdot q^{-1} \bmod 998244353$, where q^{-1} is the modular inverse of q modulo 998244353.

Input

The first line contains a single integer T ($1 \le T \le 10^5$) — the number of test cases.

The description of the test cases follows.

For each test case:

- The first line contains two integers n ($2 \le n \le 3 \times 10^5$) and q ($1 \le q \le 3 \times 10^5$) — the length of the array and the number of queries.
- The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array.
- Each of the next q lines contains four integers l_1, r_1, l_2, r_2 ($1 \le l_1 \le r_1 < l_2 \le r_2 \le n$) — describing the two non-intersecting ranges.

It is guaranteed that the sum of n over all test cases does not exceed 3×10^5, and the sum of q over all test cases does not exceed 3×10^5.

Output

For each query, output a single line containing the answer modulo 998244353.

Examples

standard input	standard output
1 4 1 1 1 1 1 1 2 3 4	2
1 5 2 1 1 2 3 4 1 1 2 2 1 3 4 5	299473306 299473306
1 10 5 1 1 2 3 4 5 8 8 8 9 1 1 2 2 1 3 4 5 1 5 6 10 1 3 4 10 1 2 3 10	155282455 217078534 217078534 217078534 217078534

Problem L. The House of Wisdom

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Lim Green`

Mazen, curious as ever, wanted to test the newly-repaired time machine for himself. But unlike Agamay and Moamen, he wasn't interested in sightseeing—he had a problem he couldn't solve on his own.

So, he locked in the coordinates: **Baghdad, Iraq**, year **850 CE**. His destination: the legendary *House of Wisdom*, where the father of algebra, **al-Khwarizmi**, was rumored to be working.

Mazen stepped into the library and found al-Khwarizmi deep in thought. Mazen presented him with the problem:

You are given two integer arrays A and B, both of length n. The array A represents initial constants in an algorithm, and array B is a dynamic buffer that starts filled with zeros: $B_i = 0$ for all $1 \le i \le n$.

The algorithm supports two types of operations:

- **Type 1**: Add a fixed value x to all elements in B from index l to r.
- **Type 2**: For each index i from l to r, define $C_i = \max(A_i, B_i)$, and output the **minimum value** among $C_l, C_{l+1}, \ldots, C_r$.

Mazen needs your help implementing this system efficiently so that he can impress the great al-Khwarizmi with a working prototype.

Input

The first line contains two integers n and q ($1 \le n, q \le 10^5$) — the size of the arrays and the number of queries.

The second line contains n integers $A_1, A_2, \ldots, A_n$ ($1 \le A_i \le 10^9$) — the elements of array A.

Each of the next q lines contains a query in one of the two formats:

- `1 l r x` ($1 \le l \le r \le n$, $1 \le x \le 10^9$)
- `2 l r` ($1 \le l \le r \le n$)

It is guaranteed that the query formats are valid.

Output

For each query of type 2, print a single line containing one integer — the minimum value of C_i in the range l to r.

Example

standard input	standard output
5 4 5 3 1 2 4 2 1 5 1 3 5 3 1 1 2 3 2 1 5	1 3

Problem M. Samurai Duel

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Bronze`

In Edo-era Japan, Mazen and Islam were on a relaxing time-travel vacation when they stumbled upon a tense local duel. Intrigued, they paused to watch, only to find out it was a battle of wits.

Mazen, fascinated by patterns, decided to challenge Islam with a mind-bending problem involving strings. The challenge was as follows:

Given two strings S and T, where S has length N and T has length M, Islam starts with an empty string U. He repeatedly performs the following operation:

- Randomly and uniformly pick an index i from 1 to N (inclusive), and append the i-th character of S to the end of U.

This process continues until the last M characters of U match the string T.

Mazen asks: what is the expected length of U when this condition is met for the first time?

Output the result modulo $10^9 + 7$.

Input

The first line contains a single integer T ($1 \le T \le 100$) — the number of test cases.

Each test case consists of three lines:

1. The first line contains two integers N and M ($1 \le N \le 2 \cdot 10^5$, $1 \le M \le 800$).
2. The second line contains the string S ($|S| = N$).
3. The third line contains the string T ($|T| = M$).

It is guaranteed that:

- The sum of N across all test cases does not exceed $2 \cdot 10^5$.
- The sum of M across all test cases does not exceed 800.
- Both S and T consist of lowercase English letters.
- All characters in T are also present in S.

Output

Output the expected length of U when the last M characters of U first match T. Print the result modulo $10^9 + 7$.

Example

standard input	standard output
3 1 3 a aaa 2 1 ab a 3 2 aab ab	3 2 500000008

Problem N. Designing the Wall

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Pink`

Zyad, a fan of architecture, set the time machine to ancient China, determined to witness the construction of the Great Wall with his own eyes.

But when he arrived, something was off — no majestic stone serpents carved across mountaintops. He had arrived too early. The Wall wasn't even conceived yet.

While wandering the region, he overheard imperial planners struggling with a peculiar challenge: *how to route the wall across jagged mountain ranges.* They needed a way to identify strong natural peaks, not just by height, but based on how they stood out relative to their neighbors. The planners had an idea — a "peak"should rise higher than nearby terrain, framed symmetrically by equal foothills.

Curious and excited, Zyad decided to help — by coding.

You are given an array a representing the heights of consecutive mountains.

For a given range $[L, R]$, a **peak** is defined as a mountain with index k satisfying the following:

- $L < k < R$;
- There exist two indices i and j such that $L \le i < k < j \le R$.
- For these two mountains $a_i = a_j$ and $a_k > a_i$.

In simpler terms, a mountain is considered a peak if it rises between two equal-height foothills within the same range.

You are given Q queries. Each query specifies a range $[L, R]$ in the array, and you must report the height of the highest peak in that subarray. If no peak exists in that range, output 0.

Input

The first line contains a single positive integer number $1 \le T \le 10^5$, the number of testcases.

The first line of each testcase contains a single positive integer number $3 \le N \le 10^5$, the length of the array a.

The next line contains the elements of the array $1 \le a_i \le 10^9$.

The next line contains a single positive integer number $1 \le Q \le 10^5$, the number of queries.

Each of the following Q lines contains two integers L and R such that $1 \le L < R \le 10^5$. It is guaranteed that $R - L > 1$.

It is guaranteed that the sum of N and Q over all testcases does not exceed 10^5.

Output

For each query in each testcase, output the answer to that query in a separate line as described in the problem statement.

Example

standard input	standard output
2 4 2 1 3 2 2 2 4 1 4 8 1 10 2 3 2 6 3 6 4 1 5 1 8 3 6 3 8	0 3 3 6 3 6