Problem A. Adding Integers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given integers n and k.

For a positive integer q, $f(q)$ is defined as the sum of $\binom{n}{a_1} \cdot \binom{a_1}{a_2} \cdot \ldots \cdot \binom{a_{q-1}}{a_q}$ for all integer sequences $(a_1, a_2, \ldots, a_q)$ that satisfy the condition $n \geq a_1 \geq a_2 \geq \ldots \geq a_q \geq 0$.

Calculate the value $\sum_{q=1}^{k} f(q)$ modulo $998\,244\,353$.

Here, $\binom{A}{B}$ is the binomial coefficient: the number of ways to select B distinct items from A distinct items.

Input

The first line of input contains an integer t: the number of test cases ($1 \leq t \leq 10^5$).

Each of the following t lines contains two integers: n and k ($0 \leq n \leq 10^9$; $1 \leq k \leq 2 \cdot 10^5$).

The total sum of k over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, print the value of the sum modulo $998\,244\,353$.

Example

standard input	*standard output*
4 2 2 0 1 271 818 141 42	13 1 812506614 405709861

Problem B. Bottles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

An immortal elf got $e + p + w$ bottles from the mage.

The first e bottles contain elixir. If the elf drinks it, she becomes immune to all poison effects, present or future.

The other p bottles, numbered from 1 to p, contain different poisons. Each poison has a delayed effect: poison bottle i has a delay of $t + i - 0.5$ days. If the elf drinks from poison bottle i, the respective delay passes, and she didn't yet drink any elixir in her life (doesn't matter if it's before or after drinking the poison), she dies. Poisons act independently: the delay for each poison is not changed by drinking other poisons.

The remaining w bottles contain water. When the elf drinks it, nothing happens.

At the same time every morning, the elf chooses one non-empty bottle with equal probability and drinks it. If all bottles are empty, she does nothing.

Find the probability that the elf will be alive $10^{10^{10}}$ days after the first day she starts drinking bottles. Remember that the elf is immortal, so she won't die from anything other than poison.

Input

The first line of input contains four integers: e, p, w, and t ($1 \le e, p, w, t \le 10^5$).

Output

It can be proven that the probability is a rational number. Represent it as p/q where p and q are coprime integers, and print the integer $p \cdot q^{-1}$ modulo 998 244 353. You may assume that q will be coprime with 998 244 353.

Examples

standard input	standard output
1 1 2 1	249561089
1 1 1 42	1
2 2 2 2	987152750

Note

For the three examples, the answers in rational form are: 3/4, 1/1, and 83/90.

Problem C. Counting Orthogonal Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

A regular polygon with n vertices has n edges and $\frac{n(n-1)}{2} - n$ diagonals. Consider the set of all these items. It contains $\frac{n(n-1)}{2}$ line segments.

Calculate how many pairs of segments from this set satisfy the following conditions:

- the segments have a common endpoint (which is a vertex of the regular polygon),
- the segments are orthogonal.

Input

The first line of input contains an integer t: the number of test cases ($3 \le t \le 10^5$).

Each of the following t lines contains an integer n: the number of vertices in the regular polygon ($3 \le n \le 10^9$).

Output

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

Example

standard input	*standard output*
3 5 4 10836006	0 4 58709502180012

Problem D. Divine Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 mebibytes

Consider a weighted tree with a gold or bronze coin placed at each vertex. Such a tree is called a *Divine Tree* if the following process is possible:

1. Zero or more times repeat the following action: select a pair of vertices that are directly connected by an edge, and swap the coins placed in those vertices.
2. Delete at most one edge from the tree. This operation may be done only once after all operations of type 1 are performed.
3. After the operation of type 2, the tree is divided into at most two trees, and for each resulting tree, the vertices contain the coins of the same metal.

The cost of each operation of type 1 is equal to the weight of the chosen edge. The cost of the Divine Tree is defined as the minimal total cost of all operations of type 1 required to transform the tree.

You are given a tree with n vertices, where n is odd. You may assume that the given tree is a Divine Tree. Let the i-th edge have weight w_i.

The tree grows in the following way: q growing events happen. In the j-th event, one of the edges e_j is chosen, and its weight is increased by d_j. The effect of growth is permanent.

Your task is to print the cost of the Divine Tree after each event.

Input

The first line of input contains an integer n: the number of vertices in the given Divine Tree ($3 \le n < 10^5$; n is odd).

The second line contains a string of length n consisting of capital letters `G` and `B`. If the i-th letter in the string is `G`, the vertex i initially contains a gold coin, and if it is `B`, the vertex contains a bronze coin.

Each of the following $n-1$ lines contains three integers, u_i, v_i, and w_i, which mean that the i-th edge connects vertices u_i and v_i and has weight w_i ($1 \le u_i, v_i \le n$; $u_i \ne v_i$; $0 \le w_i \le 10^5$). You may assume that the given graph is a Divine Tree.

Then follows a line containing an integer q: the number of events ($1 \le q \le 10^5$).

Each of the following q lines contains two integers e_j and d_j: the number of the growing edge and the weight increase, respectively ($1 \le e_j \le n-1$; $1 \le d_j \le 10^5$).

Output

Print q lines. On the j-th of these lines, print the cost of the Divine Tree after the j-th event.

Examples

standard input	*standard output*
5 BGGGG 1 3 0 2 1 0 5 2 0 2 4 0 5 2 1 1 3 4 4 3 10 1 2	0 1 1 3 5
5 GBBGB 3 2 0 2 1 0 1 4 0 1 5 1000 4 4 1 3 1 2 1 1 1	0 1 3 4
7 GGBBBBG 1 5 101 2 5 101 3 5 100 3 6 100 4 6 100 7 6 100 6 6 1 6 1 6 1 5 3 3 3 6 12345	301 302 303 303 306 711

Problem E. Eve, Adam and Three Integers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

In the arithmetic lesson, the teacher wrote three positive integers a, b, and c on the board (in that order). Adam said that there are two numbers of different parity written next to each other on the board. Eve said that there are two numbers of the same parity written next to each other on the board.

Given a, b, and c, determine whether both Adam and Eve could be right at the same time.

Input

The first line of input contains an integer a ($1 \le a \le 1000$) — the leftmost of the numbers written by the teacher on the board.

The second line of input contains an integer b ($1 \le b \le 1000$) — the middle number written by the teacher on the board.

The third line of input contains an integer c ($1 \le c \le 1000$) — the rightmost of the numbers written by the teacher on the board.

Output

Output 1 if both Adam and Eve were right, and 0 otherwise.

Examples

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

Problem F. Fibonacci Triangles

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The Fibonacci sequence is defined as follows: $f_0 = f_1 = 1$, $f_i = f_{i-1} + f_{i-2}$ for $i > 1$.

Count the number of distinct triangles with a non-zero area, all side lengths of which are elements of the Fibonacci sequence, and the largest side does not exceed the n-th element of the sequence. Triangles that can be obtained from each other by rotations and reflections are considered the same.

Input

The input contains a single integer n ($1 \le n \le 10^9$).

Output

Output a single number — the required number of triangles.

Example

standard input	standard output
2	3

Problem G. Game of Voleyball

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A volleyball match starts with a score of 0:0 in sets and 0:0 in the current set. The game consists of rallies, each rally ending with the victory of one of the teams. The winning team receives 1 point, while the losing team receives 0.

The match is played until one team wins three sets. Sets one through four are won by the first team to score 25 or more points with a lead of at least 2 points. Thus, a team can win a set with scores of 25:0, 25:1, and so on up to 25:23. A score of 25:24 does not mean the end of the set, as the point difference is not at least 2. In this case, the game continues until one of the teams achieves a point difference of at least 2 (i.e., 26:24, 27:25, 28:26, etc.). If the set score becomes 2:2, then the fifth set is played to 15 points, while the rule that a lead of at least 2 points is required to win the set remains in effect. When one team wins a set but does not win the match, a new set begins with the next rally, with both teams starting at 0 points.

Input

The input consists of a single line made up of the characters `A` and `B`, indicating which team won each rally, in the order the rallies were played.

It is guaranteed that the input is valid, meaning that if any letter represents a point that ends the match, there are no more letters following it in the input. The input string contains at least one and no more than 200 characters.

Output

In the first line of standard output, print the result by sets. If the match is not finished, then in the second line print the result in the current set (in each line, first print the result for team A, then a colon, and finally the result for team B).

Examples

standard input
`AABABAABBBAAAABABBAABABAAAABBAAABABABABABABBABBA`
standard output
`1:0` `2:4`

standard input
`AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA`
standard output
`3:0`

standard input
`ABBAAAABBBBBBAAABBBBBBABBBBBBBBBBAB`
standard output
`0:1` `0:0`

Problem H. Heroes and Illusions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

In one well-known MOBA game, n heroes are aligned in a row. However, some of them may be illusions.

The observer counted the number of real heroes from ℓ-th to r-th inclusively for every pair of ℓ and r ($1 \le \ell \le r \le n$), and recorded their evenness. Her $\frac{n(n+1)}{2}$ records show that there were k intervals that contained an odd number of real heroes. How many possible hero alignments are there? The answer may be too large, so print it modulo $998\,244\,353$.

Two alignments are considered different if, for some i, the i-th hero from the left is real in one alignment and an illusion in the other.

Input

The first line of input contains an integer t: the number of test cases ($1 \le t \le 10^5$).

Each of the following t lines contains two integers: n and k ($1 \le n \le 10^5$; $0 \le k \le n(n+1)/2$).

Output

Print the number of possible hero alignments modulo $998\,244\,353$.

Examples

standard input	*standard output*
1 5 9	10
4 3 4 6 12 6 11 12 30	3 35 0 286

Problem I. Internet Connection Stability

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

To monitor the functionality of the mobile communication network, the operator collects a large amount of internal statistics.

In particular, for each base station, the number of established connections involving subscribers located within the territory of that base station is recorded. In this problem, at the moment of establishing a connection, each subscriber is located within the territory of exactly one base station; establishing a connection increases the counters by 1 (let's call this a type 1 connection). If two subscribers are located within the territory of the same base station, then the connection counter for that station increases by 1 for each subscriber (let's call this a type 2 connection).

You are given a list of the number of connections for all n base stations, collected over the last few days. You need to determine the minimum number of type 2 connections that could have been among the connections for which information was collected.

Input

The first line of the input contains an integer n ($1 \leq n \leq 10^5$) — the number of base stations. The i-th of the following n lines contains one integer a_i ($0 \leq a_i \leq 1'000'000$) — the value of the connection counter for the i-th base station. It is guaranteed that the sum of all a_i is even.

Output

Output a single integer — the minimum number of type 2 connections that could have been among the connections that formed the given list of counter values.

Examples

standard input	standard output
5 4 3 2 1 0	0
1 4	2

Problem J. Jumping Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

The rules of the game are simple.

There is a chessboard with r rows and c columns with a chess knight on it. The cell at the i-th row from the top and the j-th column from the left is called square (i, j). Initially, the knight is placed on square (r_s, c_s).

Annapurna and Brahma alternately take the following action, starting with Annapurna:

- Move the knight onto one of the squares on the board that the knight has never visited since the beginning of the game. Remember that knights can move from (x_1, y_1) to (x_2, y_2) if and only if $(x_1 - x_2)^2 + (y_1 - y_2)^2$ is 5.

The player who cannot move the knight loses the game, and their opponent is declared the winner. Determine whether Annapurna or Brahma will win if both play optimally.

Input

The first line of input contains an integer t: the number of test cases ($2 \le t \le 2 \cdot 10^5$).

Each of the following t lines contains four integers, r, c, r_s and c_s: the number of rows and columns of the board, as well as the starting row and column for the knight, respectively ($1 \le r, c \le 10^9$; $1 \le r_s \le r$; $1 \le c_s \le c$).

Output

Output t lines. On the i-th line, print the name of the winner for the i-th test case: `Annapurna` or `Brahma`.

Example

standard input	*standard output*
2 6 6 6 6 7 19 7 3	Annapurna Brahma

Problem K. Kangaroo On Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

You are given a weighted directed graph consisting of n vertices and m edges, with vertices numbered from 1 to n and edges numbered from 1 to m. The j-th $(1 \le j \le m)$ edge goes from vertex u_j to vertex v_j $(u_j < v_j)$, and its weight is w_j.

Also, k triplets of integers are given. The i-th $(1 \le i \le k)$ triplet is (a_i, b_i, c_i) $(a_i < b_i < c_i)$.

Kangaroo starts at vertex 1 and goes to vertex n by repeatedly moving along an edge. In addition, for all i $(1 \le i \le k)$, if the kangaroo moves from vertex a_i to vertex b_i directly, then it must next move to a vertex other than vertex c_i.

Determine whether it is possible for the kangaroo to reach vertex n. If it is possible, also calculate the minimum sum of the weights of the edges on the kangaroo's path.

Input

The first line of input contains two integers n and m: the number of vertices and edges in the graph, respectively $(3 \le n \le 2 \cdot 10^5;\ 0 \le m \le 2 \cdot 10^5)$.

The j-th of the following m lines contains three integers, u_j, v_j, and w_j: the starting and the ending point of the i-th edge and its weight, respectively $(1 \le u_j < v_j \le n;\ (u_i, v_i) \ne (u_j, v_j)$ for $i \ne j;\ 1 \le w_j \le 10^9)$.

Then follows a line containing an integer k: the number of forbidden triples $(0 \le k \le 2 \cdot 10^5)$.

Each of the following k lines contains three integers: a_i, b_i, and c_i $(1 \le a_i < b_i < c_i \le n)$. You may assume that both edges (a_i, b_i) and (b_i, c_i) exist in the graph.

Output

If vertex n is unreachable, print -1. Otherwise, print the minimum sum of the weights of the edges on the kangaroo's path.

Examples

standard input	*standard output*
4 4 1 3 2 1 2 3 2 4 3 3 4 3 1 1 3 4	6
7 8 1 3 5 1 2 2 3 4 1 2 4 1 4 5 6 4 6 2 5 7 1 6 7 1 2 3 4 5 2 4 6	9
4 3 1 2 3 2 3 4 3 4 1 1 1 2 3	-1

Problem L. Lattice Sets

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

A straight line is considered to be an axis of symmetry for a figure if for every point of the figure, the point symmetric to it with respect to the line also belongs to this figure. In particular, under symmetry, boundary points map to boundary points, and interior points map to interior points.

Your task is to select a non-empty set of exactly k cells on a 1000×1000 grid such that the figure formed by these cells has as many axes of symmetry as possible. As an answer, output the number of axes of symmetry of this figure.

Input

The first line of input contains a single integer k ($1 \le k \le 1000$).

Output

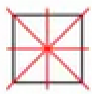
Output a single integer — the maximum number of axes of symmetry for a non-empty figure consisting of k cells.

Examples

standard input	standard output
1	4
3	2
42	2

Note

In the first test case, $k = 1$, meaning the only figure that can be obtained is a unit square.



It has 4 axes of symmetry: two lines connecting the midpoints of opposite sides, and two diagonals, so the answer is 4.

In the second test case there are two axes of symmetry, for example, for a rectangle with sides three and one.