0_Hello World

(30 points)

Introduction

YTP Contest has started!

Let’s verify everything first.

Is the internet setting correct?

Is the source code submission working well?

Do you use STDOUT output for program solutions?

Everything is ready! Go get 30 points now!! Go! Go! Go!

Statement

Please write a program to output Hello World!

Input Format

This problem requires no input.

Output Format

[A~Z][a~z], space, and common English punctuation.

Constraints

[A~Z][a~z], space, and exclamation mark "!".

Test Cases

Input 1

(no input)

Output 1

```
Hello World!
```

Illustrations

Input 1 has no input, simply output Hello World!

1_Ways of travel in n dimensions

(5 points)

Statement

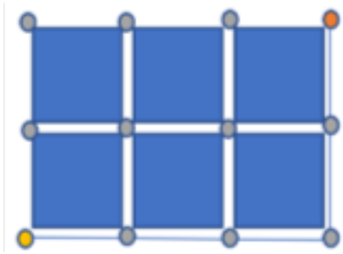
When traveling from the origin of a 2D plane, $(0,0)$, to (x,y), the traveler takes one step at a time, moving either x-axis or y-axis positively until they reach to (x,y). As result, there are $\binom{x+y}{x}$ ways to travel to (x,y) from $(0,0)$.

In detail, the notation $\binom{n}{m}$ is defined as,

$\binom{n}{m} = \frac{n!}{m! \times (n-m)!}$

For example, there are 10 ways to $(3,2)$, which can be calculated by the following,

$\binom{3+2}{3} = \binom{5}{3} = \frac{5!}{3! \times 2!} = \frac{120}{6 \times 2} = 10$



We need your help to expend this formula to calculate the ways to travel in n dimensions, such as, from $(0,0,0,0,\ldots)$ to $(a_1,a_2,a_3,a_4,\ldots,a_n)$.

Input Format

$a_1,a_2,a_3,a_4,\ldots,a_n$ is a series of integers separated by comma represents the location.

Output Format

A positive integer N indicates the total ways to reach the location.

Constraints

- Dimension n satisfies $1 \leq n < 10$.
- Inputs a_i satisfies $0 \leq a_i < 10$.
- It is guaranteed that the output value, N, satisfies $0 < N < 2^{31} - 1$.

Test Cases

Input 1

```
3,2
```

Output 1

```
10
```

Input 2

```
2,3,1
```

Output 2

```
60
```

Input 3

```
1,0,2,3,3,0,2,1
```

Output 3

```
3326400
```

Illustrations

Case 1. explained in the example of Statement.

Case 2. there are 60 ways from $(0, 0, 0)$ to $(2, 3, 1)$ in a 3D space, as shown bellow.

2_Coool Computing

(10 points)

Statement

There are many studies showing that the ternary (3-based) is much more efficient than the well-known decimal (10-based) and binary (2-based), and can represent the most values in fewer states. Therefore, in the following, we will consider a ternary computer!

Although it is a ternary computer, due to the need to interact with humans, such as the user interface, there must be a mechanism to convert the value into decimals. As convenient, we use 11 trits (`00000000000`-`12002011200`) to represent 0-99999. It is obvious that there will not be any 2 in the highest trit of the ternary notation in this range. Thus, we use 2 as a checksum to validate that the data is correct itself or in order.

In this problem, we will provide T records, which are in the range of `00000000000`-`12002011200` has been processed by the following rules. Your task is to calculate the sum of all valid numbers for all records.

There are 3 rules to set the highest trit (the 1-st digit) to 2.

- Rule 1. On $1^{st}, 4^{th}, 7^{th}, \dots, (3k+1)^{th}$ record, sum up all 11 trits and then divide by 3. Set to 2 if the remaining is 0.
- Rule 2. On $2^{nd}, 5^{th}, 8^{th}, \dots, (3k+2)^{th}$ record, sum up all 11 trits and then divide by 3. Set to 2 if the remaining is 1.
- Rule 3. On $3^{rd}, 6^{th}, 9^{th}, \dots, (3k+3)^{th}$ record, sum up all 11 trits and then divide by 3. Set to 2 if the remaining is 2.

Input format

The first line contains an integer T -- the number of records.

In the following T lines, each line contains a string to represent each record.

Output format

The sum of all the valid records in decimal. The number is less or equal to 99999000.

Constraints

- The 11-trits records are in between `0000000000` to `22222222222`.
  As described, some records are invalid and need to be excluded by the program.

Test Cases

Input 1

```
5
00000000000
00000000001
00000000002
22222222222
20000000001
```

Output 1

```
1
```

Input 2

```
9
20000000000
20000000001
20000000002
00000000010
00000000011
00000000012
00000000020
00000000021
00000000022
```

Output 2

```
36
```

Input 3

```
11
20000000120
20000000121
20000000122
22002011200
22002011200
22002011200
22002011201
22002011201
22002011201
12222222222
12222222222
```

Output 3

```
181948
```

Illustrations

Case 1.

`00000000000` => invalid, record 1 (rule 1), value 0 should be `20000000000` in 11-trits.

`00000000001` => invalid, record 2 (rule 2), value 1 should be `20000000001` in 11-trits.

`00000000002` => invalid, record 3 (rule 3), value 2 should be `20000000002` in 11-trits.

`22222222222` => invalid, record 4 (rule 1), `22222222222` is converted to `122222222222`. But that value is more than 99999 in decimal.

`20000000001` => valid (rule 2) and the value is 1.

Case 2.

`20000000000` => valid, record 1 (rule 1), value is 0.

`20000000001` => valid, record 2 (rule 2), value is 1.

`20000000002` => valid, record 3 (rule 3), value is 2.

`00000000010` => valid, record 4 (rule 1), value is 3. $(1 \times 3 + 0 = 3)$

`00000000011` => valid, record 5 (rule 2), value is 4. $(1 \times 3 + 1 = 4)$

`00000000012` => valid, record 6 (rule 3), value is 5. $(1 \times 3 + 2 = 5)$

`00000000020` => valid, record 7 (rule 1), value is 6. $(2 \times 3 + 0 = 6)$

`00000000021` => valid, record 8 (rule 2), value is 7. $(2 \times 3 + 1 = 7)$

`00000000022` => valid, record 9 (rule 3), value is 8. $(2 \times 3 + 2 = 8)$

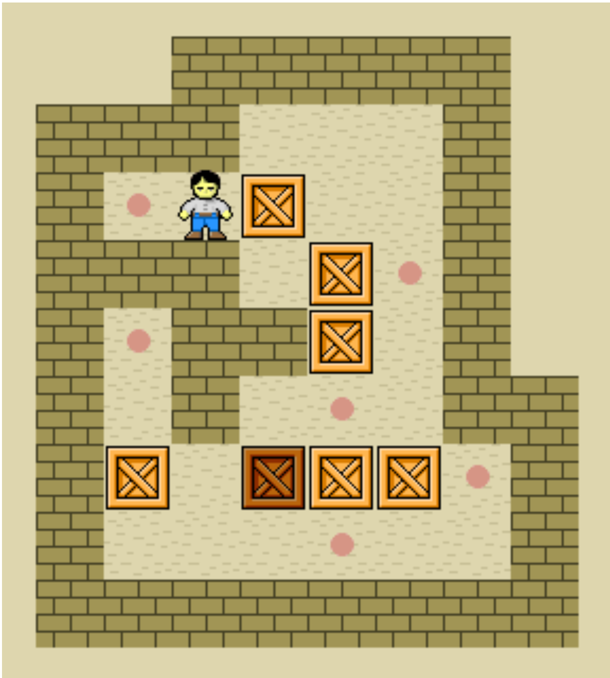
In conclusion, the total sum is 36 ($0 + 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8$).

3_Simple Sokoban

(10 points)

Statement

Sokoban is a classic video game. The player must control the man on the board by pushing the boxes up, down, left, and right. We say a board is solved if all the boxes are moved individually to any target squares. Hence, a board that is solvable indicates that the player can make a series of movements to make the board be solved.



meowmeowRanger thinks that Sokoban is too hard, so he decides to remove the man on the board and redesign the game. The new rule is: you're given a board, there're walls, several boxes, and target squares. The number of boxes is equal to the number of target squares. In each operation, you can directly move any box up, left, down, or right one square unless it hits a wall or moves out of the board. **Also, there can't have two boxes on the identical square at any moment.** Same as the original, the player should move all the boxes individually to any target squares to solve the board.

meowmeowRanger has designed several boards, but he doesn't know whether these boards are solvable. Please write a program as a validator to help meowmeowRanger verify whether each board is solvable.

Input Format

The first line contains one integer T - the number of boards meowmeowRanger has designed.

The first line of each board contains two integers, N and M, representing the board size as $N \times M$. Next, followed by N lines, each line contains M characters. The j-th character of i-th line represents the type of square (i, j) in the board, from $(1, 1)$ to (N, M). There are four types of a square. A character by the following represents each type:

- `#` : Wall
- `A` : Box

- `B` : target square
- `.` : empty square

Output Format

For each board, output a string in one line. If the board is solvable, output `YES`. Otherwise, output `NO`.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N, M \leq 1000$
- There's at least one square with type `A` in the board.
- The number of boxes is equal to the number of target squares.

Test Cases

Input 1

```
1
5 5
##.B#
..AA#
B####
#A..B
...##
```

Output 1

```
YES
```

Input 2

```
2
3 5
A.#.B
..#..
.....
3 5
A.#.B
..#..
..#..
```

Output 2

```
YES
NO
```

Input 3

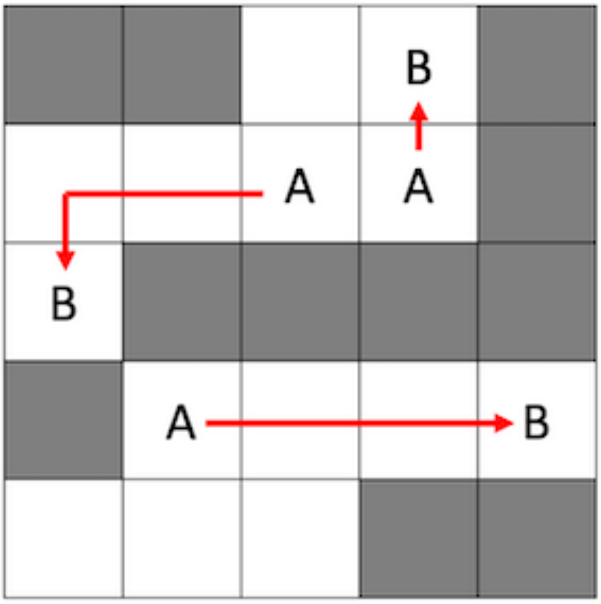
```
1
5 4
#.B.
.#..
A.#.
.#B.
.##A
```

Output 3

```
NO
```

Illustrations

- The following figure illustrates one of the solution.



- The solution of the first board in the first test case is trivial. As for the second board, we cannot move the box to the target square since there is one extra wall, the board is not solvable.
- In the third test case, we cannot move the box on the left to any of the target square because of the middle wall, so the board is not solvable.

4_Little lamb: Love Is War

(15 points)

Statement

A cute little penguin who lives in Antarctica has a crush on a lovely little lamb who lives at the North Pole. The little penguin intends to express his deep love, but the long distance keeps him from saying anything. After being aware of it, the little lamb attempts to convince him to reveal his feelings.
Knowing the number-shaped cabbage is the little penguin's favorite, the little lamb decided to plant D heads of cabbage (each head has a different shape like a number) as a love gift, implying that she is looking forward to his love.
However, the cabbage is too heavy to deliver more than four heads. In addition, the little lamb prefers the number Y, so she hopes that the sum of the four chosen numbers should be Y. How many alternatives are there for her to choose from? Would you mind helping her figure out the problem?

In short, count the numbers of 4-tuples (a, b, c, d) such that $X_a + X_b + X_c + X_d = Y$ where $a < b < c < d$.

Input Format

The first line contains two integers D, Y, representing the number of heads of cabbage and the special number.

The second line contains D integer $X_1, X_2, \ldots, X_D$, representing the i-th head of cabbage has the shape like the number X_i.

Output Format

The number of alternatives chosen from four heads of cabbage within the D heads of them that can make the sum.

Constrains

- $4 \leq D \leq 2000$
- $0 \leq Y \leq 10^8$
- $0 \leq Xi \leq 10^8$

Test Cases

Input 1

```
4 10
1 2 3 4
```

Output 1

```
1
```

Input 2

```
5 10
1 2 3 5 4
```

Output 2

```
1
```

Input 3

```
5 8
1 2 3 1 2
```

Output 3

```
2
```

Illustrations

Case One: (0 , 1 , 2 , 3) => One alternative.
Case Two: (0 , 1 , 2 , 4) => One alternative.
Case Three: (0 , 1 , 2 , 4) , (1 , 2 , 3 , 4) => Two alternatives.

5_Tommy's Dumbbell

(10 points /5 points)

Statement

Tommy has a special AiYang-dumbbell which can support as many weights as you want. We all know that a dumbbell will have one bar and some plates on both ends of the bar. AiYang-dumbbell is special because not only does it needs to make the weight equal on two sides, but it also needs to be exactly the same. For example, if one side is 5g+15g, the other side should also be 5g+15g rather than 10g+10g.

Now, Tommy has d plates and wants you to calculate how many different weights he can make.

Input Format

The first line contains a number d, representing the number of the plates.
The second line contains d numbers a_i, representing the weight of the plates.

Output Format

A number y representing how many different weights can be made.

Constraints

- $1 \leq d \leq 30000$
- $1 \leq a_i \leq 100$

Subtask

- $1 \leq d \leq 100$ (10 points)
- No extra constraints for the other 5 points

Input 1

```
4
5 5 10 10
```

Output 1

```
4
```

Input 2

```
5
3 1 4 2 5
```

Output 2

```
1
```

Input 3

```
6
3 5 8 3 9 5
```

Output 3

```
4
```

Illustrations

Example One: Two sides can be 0g, 5g, 10g, 15g
Example Two: Two sides can be 0g
Example Three: Two sides can be 0g, 3g, 5g, 8g

6_Robot

(20 points)

Statement

An N-checking robot has the following specification:

- It has N *checking slots*, where you can put marbles in or not. The robot is equipped with sensors to check if each slot has a marble inside.
- It has an inbuilt *solution*, which is its preference for which slots should have marbles in them and which shouldn't.
- It has a *submit button*, which, when pressed, will tell you the *parity* of slots such that it has a marble in the *solution* and in the corresponding slot.

You've found notes detailing M trials of the same N-checking robot by a mysterious person. Can you reconstruct a possible *solution* for that bot, or say that no such solution exists?

Input Format

In the first line, there will be two integers N, M, representing the number of *slots* the robot has, and the number of trials recorded.

In each of the next M lines, there will be a string S_i and a character c_i. S_i is a binary string of length N, where the j-th character of S_i signifies if a marble placed into the j-th slot or not. c_i is the result returned by the N-checking robot when the marbles are configured according to S_i.

Output Format

If there exists a possible solution, please output `CONSISTENT` on the first line, and then on the second line output N integers x_i, satisfying $0 \leq x_i \leq 1$, representing the *solution* of the robot. If there are multiple solutions, you may output any.

Otherwise, please output `INCONSISTENT`.

Constraints

- $1 \leq N, M \leq 500$
- $|S_i| = N$

Test Cases

Input 1

```
3 3
100 1
010 0
001 1
```

Output 1

```
CONSISTENT
1 0 1
```

Input 2

```
4 2
1010 0
0101 1
```

Output 2

```
CONSISTENT
0 1 0 0
```

Input 3

```
3 3
010 0
100 1
110 0
```

Output 3

```
INCONSISTENT
```

Illustrations

For the first test case, the singular ones test the values of the solution one by one: `101`. For the second test case, one possible solution is `0100` - the first row has no correct marbles, so the result is zero; the second row has the second marble correctly put, so has a result of one.

It can be proven that there exists no solution for the third and last testcase.

7_Candy Shop

(5 points /15 points)

Statement

meowmeowRanger loves candies, so he decided to open his own candy shop to make money. His shop has N kinds of candies numbered from 1 to N. Initially, the number of all kinds of candies is 0. The candy shop will be open for P days. On each day, meowmeowRanger will

1. replenish a consecutive interval of candies, or
2. there will be a consecutive interval of candies sold for some specific number.

For simplicity, you will receive l, r, y for each day, denoting that the number of each candy numbered from l to r changes by y. It represents replenishment if y is greater than or equal to 0 or selling off if y is less than 0. Notice that the number of candies can become negative, which means that the customers order the candies in advance.

meowmeowRanger is curious about when the number of some kinds of candies first becomes greater than or equal to k. Please write a program to help meowmeowRanger.

Input Format

The first line contains three integers N, P, Q, representing that there are N kinds of candies, the candy shop will be open for P days, and there are Q queries.
Next, followed by P lines, the i-th of which contains three integers l_i, r_i, y_i, represents that the number of each candy in the consecutive interval l_i to r_i changes by y_i on the i-th day.
Next, followed by Q lines, the i-th of which contains two integers x_i, k_i, representing a query about the first moment that the number of the candy numbered x_i is greater than or equal to k_i.

Output Format

For each query, output the desired answer, or output -1 if the number of the x_i candy will never be greater than or equal to k_i .

Constraints

- $1 \leq N, P, Q \leq 5 \times 10^5$
- $1 \leq l_i \leq r_i \leq N$
- $-10^9 \leq y_i \leq 10^9$
- $1 \leq x_i \leq N$
- $1 \leq k_i \leq 10^9$

Subtasks

- Subtask 1 satisfies $1 \leq P, Q \leq 2000$

Test Cases

Input 1

```
5 3 3
3 4 2
2 5 10
4 4 7
4 1
1 2
4 19
```

Output 1

```
1
-1
3
```

Input 2

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

Output 2

```
4
3
-1
4
```

Input 3

```
2 3 2
1 1 -10
1 2 10
2 2 -10
1 10
2 10
```

Output 3

```
-1
2
```

Illustrations

In the first case, the numbers of the fourth candy in the three days are $2, 12, 19$, respectively.
Since the number of the candy is greater than or equal to 1 on the first day, we output 1 for the first query.
Similarly, the third day is the first moment that the number of the candy is greater than or equal to 19, so we output 3 for the third query.
We output -1 for the second query since the number of the first candy never changes.

In the second case, we list the numbers of candies in the four days we may want to focus on by the following:

- The eighth candy: $3, 1, 5, 10$
- The tenth candy: $0, 0, 4, 4$
- The fourth candy: $3, 1, 1, 6$
- The second candy: $3, 3, 3, 8$
  The answers are straightforward.

Likewise, for the third one:

- The first candy: $-10, 0, 0$
- The second candy: $0, 10, 0$

8_Circle and triangle

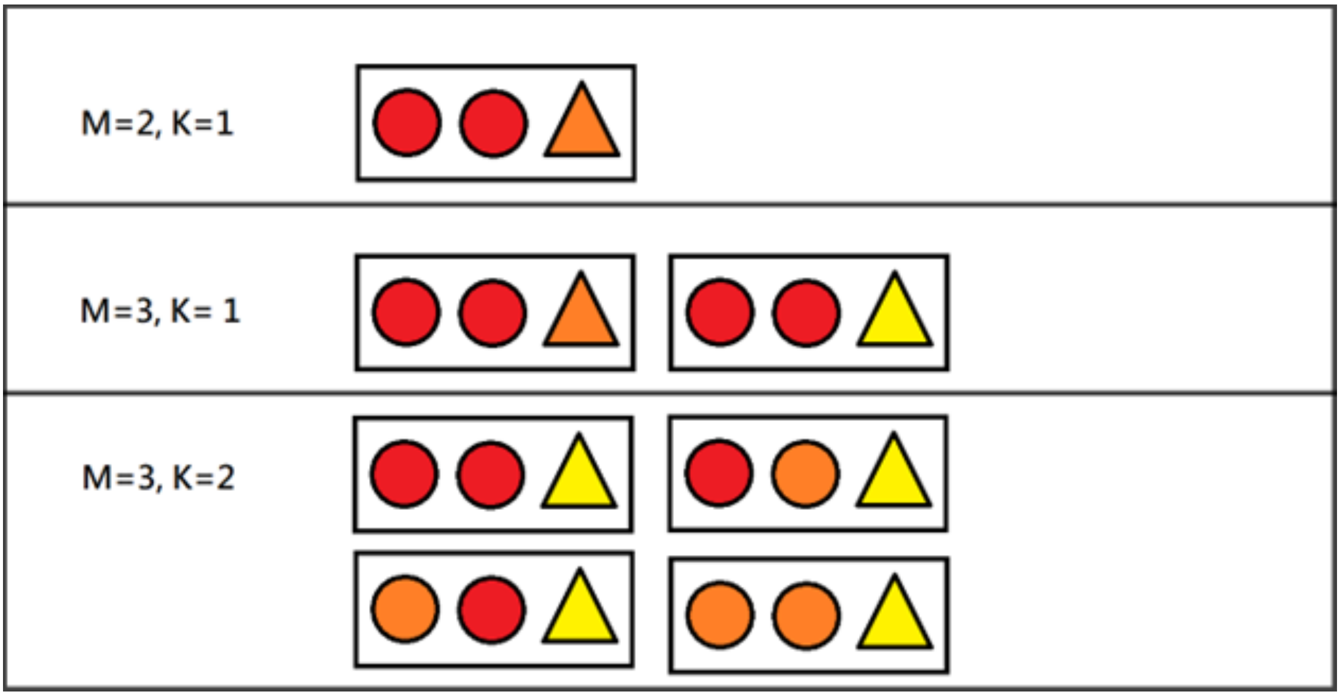
(20 points)

Statement

John has A distinct white circle blocks numbered from $1 \sim A$ and B distinct white triangle blocks numbered from $A+1 \sim A+B$.
However, John hates white. Therefore he buys M dyes to color these blocks. These dyes are numbered from $1 \sim M$, none of them is white, and they are different from each other. Each dye can be used infinitely many times.
For every circle block, John will arbitrarily use the 1-st to K-th dye to color it.
For every triangle block, John will arbitrarily use the $K+1$-th to M-th dye to color it.

However, you don't know the exact value of M and K. What you know is $2 \le M \le N$ and $1 \le K \le M-1$. In addition, N is the upper bound of M.
Two results of coloring are considered different if they have different M, or they have a difference on K, or there is a block in different colors in these two results.
Let $X(M)$ be the number of different results that can be created using M dyes.
The probability that John uses M dyes is $\frac{X(M)}{X(2)+X(3)+...+X(N)}$.

Assume that each dye costs T dollars. Please find the expective value of money that John spends on dyes. Since the answer may not be an integer, suppose it is $\frac{Y}{Z}$, please print $YZ^{-1} \mod P$.



The picture above shows all the results of $N=3, A=2, B=1$, representing the first test case, where red, orange, and yellow are the first, the second dye, and the third dye, respectively.
Though there are two results within them that have the same coloring in $M=3, K=1$ and $M=3, K=2$, they are considered different since there is a difference in K.
Besides, the blocks with the same shape are considered different, so the results on the left bottom and right top of $M=3, K=2$ are considered different.

Input Format

The first line of the input contains an integer C — the number of testcases.
Each of the next T lines contains five positive integers N, A, B, T, P — the maximum number of dyes John will buy, the number of circle blocks, the number of triangle blocks, the price of one dye, and the number P.

Output Format

For each testcases, please print a non-negative integer — the expected value of money that John spends on dyes modulo P.

Constrains

$1 \leq C \leq 10^3$
$2 \leq N \leq 10^6$, it is guaranteed the sum of N doesn't exceed 10^6
$1 \leq A, B, T \leq 10^6$
$10^8 \leq P \leq 2 \times 10^9$, it is guaranteed that P is a prime number.

Test Cases

Input 1

```
1
3 2 1 1 1000000007
```

Output 1

```
857142866
```

Input 2

```
9
2 2 2 1 1000000007
2 1 3 2 1000000009
3 1 1 3 1000000007
3 2 1 1 1000000009
3 2 2 2 998244353
3 3 1 3 998244353
3 3 2 1 1000000009
3 3 3 2 998244353
4 2 2 3 1000000009
```

Output 2

```
2
4
800000014
142857147
110916045
181498982
384615391
117440518
813953507
```

Input 3

```
1
1000000 1000000 999999 999998 998244353
```

Output 3

```
101688907
```

Illustrations

In the first case, John may buy two or three dyes.

If John buys two dyes, then John must use the first dye to color the circle blocks and use the second dye to color the triangle block. Hence, $X(2) = 1$.

If John buys three dyes, then there will be two situations:

1. John uses the first two dyes to color the circle blocks and the third dye to color the triangle block; or
2. John uses the first dye to color the circle blocks and the last two dyes to color the triangle block.

There are $2^2 \times 1^1 = 4$ different results in the former case, and $2^1 \times 1^2 = 2$ different results in the latter case. Hence $X(3) = 4 + 2 = 6$.

Therefore, the expective value of money John spends is $\frac{2 \times X(2) + 3 \times X(3)}{X(2) + X(3)} = \frac{20}{7}$.
$20 \times 7^{-1} \mod P = 857142866$. Hence you should print 857142866.

9_Tree

(20 points)

Statement

This is an interactive problem.

Mr. T is a professional spy. One day, he tried to sneak into the enemy's castle to steal some secret information. However, he was too careless to leave his daughter Y alone at home. As he finally got into the castle, he was informed through the telegraph that the enemy's spy kidnaped Y as a hostage. Despite the unfavorable situation, Mr. T had come up with a solution that required Y's mind-reading power. Mr. T planned to use Y's mind-reading power from a distance and send her the castle map. Since Y is still a young girl, she only recognizes binary numbers. Can you, as a smart guy, help Mr. T and Y come up with a protocol to transport this map?

To solve this problem:

- You should implement two functions `send`, `recv`
- `send` represents Mr. T's strategy. The function will receive a rooted tree as a parameter. You should pass a binary code as short as possible to Y.
- `recv` Will transform the binary code received from `send` into a rooted tree.
- Note: Your processes will be run separately. You will not be able to communicate with any resources other than the binary code. (ex: global variables)

Input and Output

Function interface:

```
std::string send(int N, std::vector<std::pair<int, int>> edges);
```

- `N` represents the size of the castle, which is the number of nodes
- `edges` consists `N-1` `pair<int,int>` , `edges[i]` is the ith edge connecting u_i, v_i
- please return a string constructed by `'0'` and `'1'`
- The tree is rooted by 0

```
std::vector<std::pair<int, int>> recv(std::string code);
```

- `code` a string constructed by `'0'` and `'1'` received from `send`
- please return a `vector<pair<int,int>>` of size `N-1` , the ith `pair` is the i-th edge connecting u_i, v_i

Scoring

- First, the tree returned from `recv` should be isomorphic to the tree `send` got
- The code returned from `send` should only consist `'0'` or `'1'`
- The **sum** of the time consumed by two processes should not exceed the time limit

We say two trees A and B are isomorphic if there exists an ordering of the nodes from A (except the root) such that every edge (u, v) on B should also exist on the reordered tree A.

For each test, if there are N nodes and the output has length L the score equals:

- $max(min(100, \frac{46N-L}{44N} \times 70 + 30), min(30, \frac{1.5N^2-L}{N^2} \times 30), 0)/5$

The total score for this problem is the average of each test.

Constraints

- $1 \leq N \leq 200,000$
- $0 \leq u_i, v_i < N$

Test Cases

Compile and run by using the scripts and case grader from CMS.

```
chmod +x compile_cpp.sh
./compile_cpp.sh
./Tree
```

Note: The testing script is different from the one on CMS. It does not separate the two processes.

Input 1

```
3
0 1
0 2
```

Output 1

```
3
0 2
0 1
```

Input 2

```
5
0 1
0 2
2 3
2 4
```

Output 2

```
5
0 3
0 4
4 1
4 2
```

Input 3

```
1
```

Output 3

```
1
```

Hint

- Although the outputs from the first two cases are not exactly the same, they are isomorphic trees rooted by 0. It will be considered correct.
- There may be more than one correct output.

10_Pika ☆ Pika ☆ Starry Sky Festival

(1 points /2 points /3 points /4 points /5 points /10 points)

Statement

Pika ☆ Pika ☆ Starry Sky Festival, the annual school festival of the starry sky elementary school, is around the corner. Three students of the baking club, Sara, Sena, and Matsuri, are designing a splendid cake for the festival under the supervision of Teacher Pinky.

The cake will be a rectangular prism with a height of one, and the batter they prepared is just enough for a cake of volume N. In the baking classroom, there are two rulers. One of the rulers has A marks on it, and the distance between two marks is 1; the other has B marks on it, and the distance between two marks is $\sqrt{86}$ (maybe Teacher Pinky had just finished watching the anime "86 -Eighty Six-"?). Using these two rulers together, they can measure any length of the form $a + b\sqrt{d}, |a| < A, |b| < B$. They are planning to use these two rulers to measure the length and width of the mold for the cake.

Please help them and tell them the options they have on the size of the cake.

In short, given N, A, B, please output all the integer quadruples a, b, x, y in lexicographical order such that

1. $(a + b\sqrt{86})(x + y\sqrt{86}) = N$
2. $0 \leq a + b\sqrt{86}, x + y\sqrt{86}$
3. $|a|, |x| < A$
4. $|b|, |y| < B$

Input Format

The input consists of one line, which contains three integers N, A, and B separated by a single space.

Output Format

On the first line, please output an integer t, denoting the number of options on the size of the cake.
On each of the following t lines, please output four integers a, b, x, y, separated by a single space, describing a possible size $a + b\sqrt{86} \times x + y\sqrt{86}$ of the cake.

Please order the sizes in lexicographical order with respect to the quadruple (a, b, x, y).

Constrains

- $1 \leq N \leq 10^9$
- $1 \leq A, B \leq 10^{18}$

Subtasks

- $B = 1$ (2%)
- $A, B \leq 20$ (3%)
- $A, B \leq 2000$ (15%)
- $N = 1$ (20%)
- $N \leq 1000$ (20%)

- No extra constraints (40%)

Test Cases

Input 1

```
10 20 20
```

Output 1

```
8
-18 2 9 1
-9 1 18 2
1 0 10 0
2 0 5 0
5 0 2 0
9 1 -18 2
10 0 1 0
18 2 -9 1
```

Input 2

```
86 43 43
```

Output 2

```
1
0 1 0 1
```

Input 3

```
84 200 10
```

Output 3

```
20
1 0 84 0
2 0 42 0
3 0 28 0
4 0 21 0
6 0 14 0
7 0 12 0
10 -1 60 6
10 1 60 -6
```

```
12 0 7 0
14 0 6 0
20 -2 30 3
20 2 30 -3
21 0 4 0
28 0 3 0
30 -3 20 2
30 3 20 -2
42 0 2 0
60 -6 10 1
60 6 10 -1
84 0 1 0
```

A hint that might be useful

$\mathbb{Z}[\sqrt{86}] = \{a + b\sqrt{86} | a, b \in \mathbb{Z}\}$ is a unique factorization domain. That is, every number in it can be represented as a "unique" product of "primes" in it.

In other words, there won't be cases like $(3 + \sqrt{5})(3 - \sqrt{5}) = 4 = 2 \cdot 2$ where there are two different factorizations.

Note that two prime factorizations $n = p_1 \dots p_k = q_1 \dots q_l$ are considered identical if and only if $k = l$ and there exists a permutation a_i such that $p_{a_i} | q_i$ and $q_i | p_{a_i}$.
For example, $6 = -2 \cdot -3 = 3 \cdot 2$ are identical factorizations, and $9 = 3 \cdot 3 = (9 + 6\sqrt{2})(9 - 6\sqrt{2})$ are also identical factorizations: $3/(9 + 6\sqrt{2}) = 3 - 2\sqrt{2}$ is an "integer", and so is $(9 + 6\sqrt{2})/3 = 3 + 2\sqrt{2}$.

11_Human Resource Allocation

(4 points /6 points /7 points /8 points)

Statement

It has been busy for the YTP (Yummy TeaPot) company recently. To solve business at the lowest cost, the YTP company hired several forecasting experts to analyze the N tasks they will receive in the next N days. Each task has an ability interval $[l, r]$ and the penalty money t. Also, it is necessary to find a "helper" to solve it. Assume that the helper of this task has the ability p, then,

- If $p < l$ or $p > r$, it needs to cost t dollars.
- Otherwise, if $l \le p \le r$, it won't need to cost any dollar, but the helper's ability will turn into $l + r - p$ after finishing the task.

Notice that a helper can always solve any task whatever his or her ability is. There will only be a difference in cost.

Simultaneously, the forecasting experts also predicted that each of the next N days would appear a new "helper". These N helpers have their initial ability p_i respectively. It is also expensive to hire these helpers. If the YTP company hired the i-th helper, it would need to pay c_i dollars to him or her. And once a helper is fired by the company, he or she will never come back. Otherwise, if there is a helper in the company, the task on that day must be solved by him or her. Since these helpers hate each other, if there are two or more helpers in the company simultaneously, they'll fight, and it'll cost the company heavily.

In other words, for the helper appearing on the i-th day, the YTP company needs to decide whether to spend c_i to hire this helper. If yes, then the YTP company needs to fire the helper in the company currently and start to let the new helper handle the i-th day's task. Otherwise, the helper who remains in the company will keep handling the i-th day's task without any additional money for hiring him or her.

In case of finishing all the tasks, can you help the YTP company estimate their minimum cost?

Note that you must have a helper for the first day, or there will be no helper for the first day's task.

Input Format

The first line of the input contains a positive integer N, indicating that there will be a task for each of the next N days.

The second line of the input contains N non-negative integers $p_1, \ldots, p_N$, indicating that the helper who appears in the i-th day has the ability p_i.

The third line of the input contains N non-negative integers, indicating that the helper who appears in the i-th day has the cost c_i.

Lastly, followed by N line, the i-th line has three non-negative integers l_i, r_i, t_i, indicating that the i-th task has the ability interval $[l_i, r_i]$ with the penalty money t_i.

Output Format

Output a non-negative integer in one line, indicating the minimum cost of the YTP company in case of finishing all the tasks.

Constraints

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq t_i, p_i, c_i \leq 10^9$
- $0 \leq l_i \leq r_i \leq 10^9$

Subtasks

- Subtask 1 satisfies that $N \leq 18$.
- Subtask 2 satisfies that $N \leq 3000$.
- Subtask 3 satisfies that $l_i = r_i$.
- Subtask 4 has no additional constraint.

Test Cases

Input 1

```
3
1 2 5
1 4 1
1 3 3
1 2 1
5 8 4
```

Output 1

```
3
```

Input 2

```
3
1 3 7
4 2 10
1 100 40
2 4 10
100 110 20
```

Output 2

```
14
```

Input 3

```
5
3 1 4 1 5
2 7 0 8 1
3 3 5
1 1 2
4 4 7
6 6 3
5 5 8
```

Output 3

```
8
```

Illustrations

In the first case, the best approach is as follow,

- Hire the helper on the first day and cost 1 dollar. Since the first helper has the ability 1, which is located in the interval $[1,3]$, it doesn't need any additional cost to finish the first day's task. But the ability of the first helper will turn into $1+3-1=3$.
- Don't hire the helper on the second day, but directly let the first helper continue handling the task. Notice that since the first helper's ability is 3, and the cost is 1. Even though it doesn't cost anything for the second helper with the second day's task, it costs 4 to hire him, which is less cost-effective.
- Fire the first helper on the third day. Hire the helper on the third day and cost 1 dollar. Since the third helper has the ability 5, which is located in the interval $[5,8]$, it doesn't need any additional cost to finish the third day's task. But the ability of the third helper will turn into $5+8-5=8$.

The total cost is $1+1+1=3$ dollars, which is minimum.

In the second case, the best approach is to hire the first helper to handle all the tasks, with a total cost of 14 dollars. Notice that even though at the time of the second day, the cost of the first helper is significantly higher than the second helper, if we hire the second helper at that time, it will make the cost of the third day higher, making the total cost higher.

In the third case, the best approach is to hire the first, third, and fifth helpers, respectively, with a total cost of 8 dollars. Notice that the case satisfies the constraint of subtask 3.

12_Meow (How many different breeds of cats)

(25 points)

Statement

Today, you came to a cat village. There are N cats in this village.
Each cat is very cute, and all the N cats are very cleverly lined up in a row.

As we all know, there are B breeds of cats in the world.
Breed of each cat from left to right in this village is $a_1, a_2, \ldots, a_N$ respectively.

Because you are a coder and cat lover who is very interested in cat breeds,
you may be curious about the following type of question:
given K intervals, $[l_1, r_1], [l_2, r_2], \ldots, [l_K, r_K]$,
how many different breeds of cats which appear a positive even number times in the union (that is, if a cat appears in multiple intervals, it should be counted only once) of the K intervals?

Now, give you M such questions, can you answer all correctly?

Input Format

The first line contains three positive integers
N, B and M --- the number of cats in this village, the number of breeds of cats in the world and the number of questions you need to answer.

The second line contains N positive integers $a_1, a_2, \cdots, a_N$.

The following M lines contain questions, one per line.
The first integer in the line is K, followed by $2 \times K$ integers
$l_1, r_1, l_2, r_2, \cdots, l_K, r_K$ in the same line.

Output Format

Print M lines. The i-th line contains one integer ---
the number of breeds of cats which appear a positive even number times in the given intervals.

Constraints

- $1 \leq N, B, M \leq 10^5$
- $1 \leq a_i \leq B$
- $1 \leq K \leq 10^5$
- $1 \leq l_i, r_i \leq N$
- It is guaranteed that the sum of all K's is not greater than 10^5.

Test Cases

Input 1

```
5 3 5
1 3 2 1 2
1 1 2
1 1 3
1 1 4
2 1 2 4 5
3 1 1 3 3 5 5
```

Output 1

```
0
0
1
1
1
```

Illustrations

There are 5 cats of three breeds (1, 3, 2, 1 and 2) in this sample. An explanation of each question is shown below:

1. The first question contains 1 interval - $[1, 2]$. All the breeds of cats in the union of all intervals from left to right is 1 and 3. Since no breeds of cats appear even times, the answer is 0.
2. The second question contains one interval - $[1, 3]$. All the breeds of cats in the union of all intervals from left to right are 1, 3, and 2. Since no breeds of cats appear even times, the answer is 0.
3. The third question contains one interval - $[1, 4]$. All the breeds of cats in the union of all intervals from left to right are 1, 3, 2, and 1. Since only one breed (1) appears even times, the answer is 1.
4. The forth question contains two intervals - $[1, 2]$ and $[4, 5]$. All the breeds of cats in the union of all intervals from left to right are 1, 3, 1, and 2. Since only one breed (1) appears even times, the answer is 1.
5. The fifth question contains three intervals - $[1, 1]$, $[3, 3]$ and $[5, 5]$. All the breeds of cats in the union of all intervals from left to right are 1, 2, and 2. Since only one breed (1) appears even times, the answer is 1.