2019 KAIST RUN Spring Contest

Sponsored By:

Jane Street

Naver D2

Startlink

Rules

- This contest is 2019 KAIST RUN Spring Contest.
- This contest is sponsored by Jane Street, Naver D2, and Startlink.
- You can only participate individually.
- Use of the network is prohibited during the competition, except for submitting source codes and accessing language reference sites, and using translation. Here are the allowed reference and translation sites.

```
– C/C++           : https://en.cppreference.com/w/
– Java            : https://docs.oracle.com/javase/8/docs/api/
– Python          : https://docs.python.org/
– Kotlin          : https://kotlinlang.org/docs/reference/
– Naver Papago    : https://papago.naver.com/
– Google Translate: https://translate.google.com/
```

- Using any pre-written code or prints before the contest is NOT allowed.
- The contest lasts 4 hours.
- The contest consists of 9 problems.
- Problem is ordered by the difficulty of getting full points, to our best effort. You are strongly encouraged to read all the problems.
- Each problem consists of 1 or more subtasks, and each subtask is worth a certain number of points. Subtask is an easier problem with possible additional constraints.
- Each subtask consists of several test data, and points for the subtask are awarded if every test data is passed. The total number of points for each problem is 100 points.
- Each problem have a time limit and memory limit. This means your problem should run in a given time and memory limit for each data.
- Every problem is guaranteed to be solvable using C++17 in a given time limit and memory limit. It may not be possible to solve the problem in other languages, e. g. Python.
- Memory limit for every problem is fixed as follows.:

```
– C11    / C++17   : 1024MB
– Java   / Kotlin  : 1536MB
– PyPy 3 / Python 3: 2048MB
```

- Each participant's ranking is determined in the following way.
  - Penalty time = (Duration from contest start to the last submission that increased points)
  - Ranking = (# of participants with higher points) + (# of participants with same points and lower penalty time) + 1

Contest Environment

- When you enter contest page, you can use following menus from left:
  - 메인(Main page): You can see problems of the contests. You can access problem page by clicking the title of problem.
  - 스코어보드(Scoreboard): You can see score and latest submission time for each problem, and each contestant.
  - 공지사항(Announcement): You can see announcement from the tab.
  - 질문(Question): You can ask, and see answer about question. Question related to problem should be done here.
- In problem page, you can read the problem statement. You can submit the solution using 제출 (Submit) tab.
- In submit tab, choose your programming language in 언어(Language), and paste source code to 소스 코드(source code) area. You can finally submit your solution by clicking 제출(Submit) button.
- After you click the submit button, you will be redirected to the verdict page. You can see your previous submission. Each column of table is as follows:
  - 채점 번호(Run ID): Index of your submission. Not relevant.
  - 아이디(ID): Your ID of Baekjoon Online Judge.
  - 문제 번호(Problem number): The number of problem you submitted.
  - 결과(Verdict): your solution is given as one of following verdicts:
    - 기다리는 중 (Waiting): Your solution is in judging queue, waiting for evaluation.
    - –점 (– Points): Your solution got – points in total.
    - 틀렸습니다 (Wrong Answer): Your solution got 0 point, and in one or more test data, your program terminated normally in time and memory limit, but your output is not correct.
    - 시간 초과 (Time limit exceeded): Your solution got 0 point, and in one or more test data, your program ran after time limit of problem, so program was killed. Notice that you can get time limit verdict by infinite loop in program, or program terminates but is slow. Time usage includes virtual machine start time, input, output, system call, and your processing time.
    - 메모리 초과 (Memory limit exceeded): Your solution got 0 point, and in one or more test data, your program allocated memory more than memory limit. Memory usage includes allocated stack memory and heap memory.
    - 출력 초과 (Output limit exceeded): Your solution got 0 point, and in one or more test data, your program printed too much to output stream.
    - 런타임 에러 (Runtime Error): Your solution got 0 point, and in one or more test data, your program terminated abnormally during program. Your program can get runtime error including following:
      - by accessing files or streams other than standard input, output stream(`stdin`, `stderr`),
      - by printing any content to standard error stream(`stderr`),
      - by calling system-related functions, (including `system` in C/C++, `Runtime .getRuntime().exec` in Java/Kotlin, and `system` in Python but not limited to)
      - when return code is not zero, (return code of main should be 0 in C/C++, when you terminate program using library such as 'exit' in C/C++, `System.exit` in Java, `exitProcess` in Kotlin, or `sys.exit` in Python, argument passed to it should be 0 to not get a Runtime Error)

        · by unhandled custom error, exception, or segmentation fault,
        · by creating another thread and so on.

    ∗ 컴파일 에러 (Compile Error): Your solution got 0 point, and compiler did not produced executable file.
    ∗ 메모리 (Memory used): Maximum memory used throughout inputs of all test data if you got full mark.
    ∗ 시간 (Time used): Maximum time used throughout inputs of all test data if you got full mark.
    ∗ 언어 (Language): Programming language of submission. You can see submission detail by clicking it. You can click 수정(Modify) button do modify the submission. In submission detail, you can see verdict on each subtask. The table consists of 서브태스크 번호(the number of subtask), 배점 (points assigned to subtask), 결과(verdict, same as above, but includes 맞았습니다!!(Accepted), if you passed in all test data.), 메모리(maximum memory used in subtask), and 시간(maximum time used in subtask).
    ∗ 코드 길이 (Code length): Length of submitted code.
    ∗ 제출한 시간 (Submitted time): Time you submitted your solution.

- Your score to the problem is maximum score of the submission you submitted in problem. You can see this in the scoreboard.
- Your submission runs on following environment:
  - Amazon AWS EC2
  - Instance Type: c4.large
  - Processor: Intel Xeon ES-2666v3
  - Clock: 2.9GHz
  - Memory: 3.75GiB
  - Processor Architecture: 64-bit
  - OS Ubuntu 16.04.6 LTS

Language Guide

- You can choose your programming language from following:
  - C11
    - Compile: `gcc Main.c -o Main -O2 -Wall -lm -static -std=c11 -DONLINE_JUDGE -DBOJ`
    - Execution: `./Main`
    - Language version: `gcc (GCC) 7.3.0`
  - C++17
    - Compile: `g++ Main.cc -o Main -O2 -Wall -lm -static -std=gnu++17 -DONLINE _JUDGE -DBOJ`
    - Execution: `./Main`
    - Language version: `gcc (GCC) 7.3.0`
  - Java
    - Compile: `javac -J-Xms1024m -J-Xmx1024m -J-Xss512m -encoding UTF-8 Main.java`
    - Execution: `java -Xms1024m -Xmx1024m -Xss512m -Dfile.encoding=UTF-8 Main`
    - Language version: `Java(TM) SE Runtime Environment (build 1.8.0_191-b12)`
  - Kotlin
    - Compile: `kotlinc-jvm -J-Xms1024m -J-Xmx1024m -J-Xss512m -include-runtime -d Main.jar Main.kt`
    - Execution: `java -Xms1024m -Xmx1024m -Xss512m -jar Main.jar`
    - Language version: `Kotlin version 1.3.10 (JRE 1.8.0_191-b12)`
  - Python 3
    - Compile: `python3 -c "import py_compile; py_compile.compile(r'Main.py')"`
    - Execution: `python3 Main.py`
    - Language version: `Python 3.7.1`
  - PyPy 3
    - Compile: `python3 -c "import py_compile; py_compile.compile(r'Main.py')"`
    - Execution: `python3 Main.py`
    - Language version: `PyPy 6.0.0 with GCC 6.2.0 20160901 (Python 3.5.3)`
- C11 and C++17 are different languages. Make sure you choose appropriate language when you submit your submission.
- In Java, your class name which include ‘main‘ method should be ‘Main‘.
- The difference of Python 3 and Pypy 3 is interpreter, which can vary in performance. Pypy is generally faster, but not always.
- Followings are sample code which reads to space-separated integer from standard input, and prints their sum to standard output.
  - C11

```
#include <stdio.h>

int main() {
    int a, b;
    scanf("%d %d",&a,&b);
    printf("%d\n",a+b);
    return 0;
}
```

– C++17
```
#include <iostream>
using namespace std;

int main() {
    auto a=0, b=0;
    cin >> a >> b;
    cout << a+b << endl;
    return 0;
}
```

– Java
```
import java.util.*;

public class Main{
    public static void main(String args[]){
        Scanner sc = new Scanner(System.in);
        int a, b;
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println(a + b);
    }
}
```

– Kotlin
```
import java.util.Scanner

fun main(args: Array<String>) {
    val sc: Scanner = Scanner(System.‘in‘)
    var a = sc.nextInt()
    var b = sc.nextInt()
    println(a+b)
}
```

– Python 3
```
a, b = map(int, input().split())
print(a+b)
```

– PyPy 3
```
import sys

def main():
    a, b = map(int, sys.stdin.readline().split())
    print(a+b)

if __name__ == '__main__':
    main()
```

Problem list

#	Problem Name	Time limit (All languages)	Score	Subtask score				
			Full score	1	2	3	4	5
A	Rainbow Beads	1 second	100	25	25	20	30	
B	Gosu	1 second	100	40	60			
C	Voronoi Diagram Again	5 seconds	100	15	35	50		
D	A Plus Equals B	1 second	100	36	64			
E	Water Knows The Answers	3 seconds	100	18	19	34	14	15
F	Eat Economically	3 seconds	100	16	33	51		
G	Increasing Sequence	3 seconds	100	22	17	61		
H	Jealous Teachers	3 seconds	100	32	68			
I	Dijkstra Is Playing At My House	3 seconds	100	31	24	45		

Problem A. Rainbow Beads

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

Jaehyun has a bead which consists of N jewels arranged from left to right. Each jewel is in one of the three colors: Red, Blue, and Violet, which is represented as a single character `R`, `B`, `V`. As one of the committees in an important contest, Jaehyun wants to use it as a souvenir for some participant.

Jaehyun likes a bead with diverse colors, so he defines a bead *beautiful* if every adjacent jewel has different colors. For example, `RBVBV` is a beautiful bead because every adjacent jewel has a different color. `V` is a beautiful bead because it does not have adjacent pairs. However, `RBBV` is not a beautiful bead, because two `B`s in the middle are adjacent in the string.

Not only Jaehyun likes a bead with diverse colors, but he likes a contest with diversity. This time, Jaehyun wants to make a bead that is also colorful to colorblind people. For convenience, we will only consider three kinds of people in this problem.

- Non-colorblind people, who can tell all three colors.
- Red-colorblind people (*Protanopia*), who can't tell apart red and violet: They consider violet jewels as red jewels.
- Blue-colorblind people (*Tritanopia*), who can't tell apart blue and violet: They consider violet jewels as blue jewels.

In this case, the string `RVB` is colorful for non-colorblind people, but it is not colorful for red-colorblind people as red and violet jewels are adjacent, and it is also not colorful for blue-colorblind people as violet and blue jewels are adjacent.

Jaehyun wants to pick some contiguous part of the bead and cut it out to give as a souvenir. The part Jaehyun cuts should be colorful to all three kinds of people. Note that, if the whole bead is beautiful, then Jaehyun does not necessarily cut it out, but just give the whole bead. What is the length of the longest bead he can give?

Input

The first line contains an integer N, denoting the length of the bead.

The next line contains string of length N, where every character is either `R`, `B`, or `V`.

Output

Print the maximum possible length of contiguous beads, which is colorful for all three kinds of people.

Constraints

- $1 \leq N \leq 250\,000$

Subtask 1 (25 points)

This subtask has additional constraints:

- $N \leq 100$
- There is no violet jewel. Thus, every character in string is either `R` or `B`.

Subtask 2 (25 points)

This subtask has an additional constraint:

- $N \leq 100$

Subtask 3 (20 points)

This subtask has an additional constraint:

- $N \leq 5\,000$

Subtask 4 (30 points)

This subtask has no additional constraints.

Examples

standard input	standard output

`4`
`VRRB`
	`2`

`5`
`RBBRR`
	`2`

Problem B. Gosu

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

Ho is an expert in martial arts called *Taebo*. She runs a Taebo school, and there are N students in her school. Since Ho is too old to teach Taebo, she is going to hand over her school to one of her students. To find a suitable candidate, Ho made all $\frac{N(N-1)}{2}$ pairs of students do a Taebo matchup with each other, and wrote all the results. In a Taebo matchup, exactly one person wins the match, and another person loses the match. Ho thinks that a student is good enough to receive her school if the student is a **Gosu** of Taebo.

Gosu is a Korean word which means a person who is very good at games, sports, competitive programming, etc. In Taebo, Gosu has a different meaning.

Let's define a **winning path** from player x to player y as a sequence of $K+1$ integers $a_0 = x,\ a_1,\ \cdots,\ a_K = y$, where student a_i has won student a_{i+1} for all $0 \le i < K$. We call K as the **length** of this winning path. For example, if there exists a *winning path* of length 1, we can immediately know that x has won student y. If there exists a winning path of length 2, then x may not win y directly, but there exists some other player z that x has won, and has won y.

The **distance** $d(x,\ y)$ is defined as a minimum length of winning path from x to y, if such exists. There could be a case that x can not find a winning path to y. In that case, we define $d(x,\ y) = 9000$. Note that, the path can have zero length, thus $d(i,\ i)$ is always 0.

Ho wants her student to be strong to all kind of opponents, so she defines the **weakness** of student i, as a maximum value among $d(i,\ 1),\ d(i,\ 2),\ \cdots,\ d(i,\ N)$. A student i is a **Gosu** in Taebo when the weakness of student i is minimum among all weakness values. By this definition, there can be multiple Gosu-s.

Since Ho is too old to tell who is Gosu, your task is to find a Gosu and weakness value of Gosu to help Ho. If there exist multiple Gosu-s, you can print any of them.

Input

In the first line, the number of students N is given.

In the i-th line of next N lines, a string s_i consists of `W`, `L`, and `-`. Let's denote j-th character of s_i as $s_{i,j}$. $s_{i,j}$ is given as follows:

- $s_{i,j} =$ `-`, if $i = j$.
- $s_{i,j} =$ `W`, if student i won student j.
- $s_{i,j} =$ `L`, if student j won student i.

Output

Print two space-separated integers, d and u, where student u is Gosu, and d is weakness of student u.

If there are multiple answers, you can print any of them.

Constraints

- $2 \le N \le 3\,000$
- $s_{i,i} =$ `-` $(1 \le i \le N)$
- If $i \ne j$, then $s_{i,j} =$ `W` or $s_{i,j} =$ `L`. $(1 \le i \le N)$

- If $s_{i,j} = \texttt{W}$, then $s_{j,i} = \texttt{L}$. $(1 \le i,\ j \le N)$
- If $s_{i,j} = \texttt{L}$, then $s_{j,i} = \texttt{W}$. $(1 \le i,\ j \le N)$

Subtask 1 (40 points)

This subtask has an additional constraint:

- $N \le 100$

Subtask 2 (60 points)

This subtask has no additional constraints.

Examples

standard input	standard output

`2`
`-W`
`L-`
	`1 1`

`3`
`-LW`
`W-L`
`LW-`
	`2 1`

`5`
`-WLLW`
`L-LLW`
`WW-LL`
`WWW-W`
`LLWL-`
	`1 4`

Problem C. Voronoi Diagram Again

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds

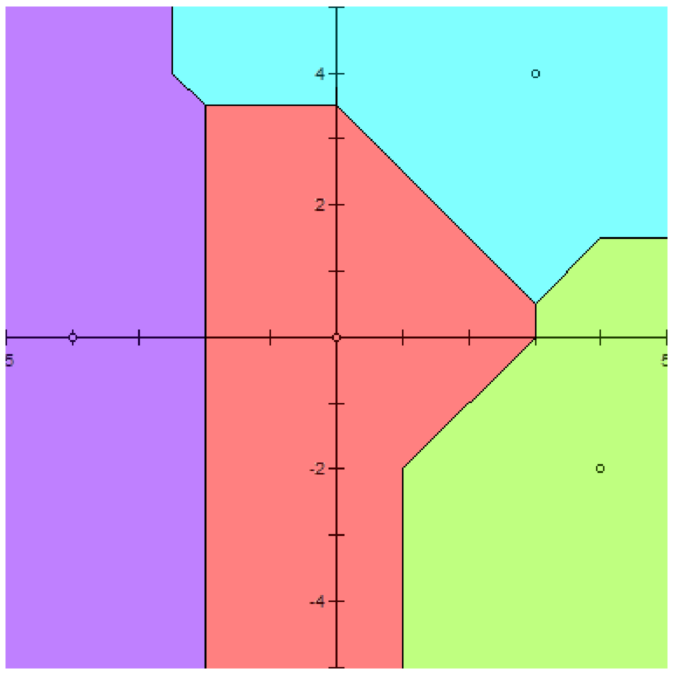


Figure: *Voronoi Diagram of size 4.*

In the 2-dimensional Cartesian coordinate system, we define the **Voronoi Diagram** of a non-empty set of points S, as a diagram that divides the plane by the criteria "which point in the set S is closest in this location?". More precisely, the Voronoi diagram of a given non-empty point set $\{P_1, P_2, \cdots, P_n\}$ is a collection of **regions**: A point K is included in region i if and only if $d(P_i, K) \leq d(P_j, K)$ holds for all $1 \leq j \leq n$.

While the usual Voronoi Diagram uses Euclidean distance, we use Manhattan distance in this problem. $d(X, Y)$ denotes the **Manhattan** distance between point X and Y. Manhattan distance between two points is the sum of the absolute differences of their X, Y coordinates. Thus, the Manhattan distance between two points (X_1, Y_1), (X_2, Y_2) can be written as $|X_2 - X_1| + |Y_2 - Y_1|$.

For example, in the picture above, every location over the plane is colored by the closest point with such location. The points which belongs to a single region is colored by a light color indicating a region, and the points which belongs to more than one region forms lines and points colored black.

The region is unbounded if for any real number R, there exists point P in the region such that $d(O, P) > R$ where O is the origin. You have to find the number of unbounded regions in the Voronoi Diagram.

Input

In the first line, the number of points consisting Voronoi diagram N is given.

In the i-th line of next N lines, two integers x_i, y_i indicating x and y coordinate of P_i are given. These are the points in the Voronoi diagram.

Output

Print a single integer, denoting the number of unbounded regions in the Voronoi Diagram.

Constraints

- $1 \le N \le 250\,000$
- $-10^9 \le x_i,\ y_i \le 10^9$ $(1 \le i \le N)$
- All N points are distinct.

Subtask 1 (15 points)

This subtask has additional constraints:

- $N \le 2\,000$
- $-100 \le x_i,\ y_i \le 100$ $(1 \le i \le N)$

Subtask 2 (35 points)

This subtask has an additional constraint:

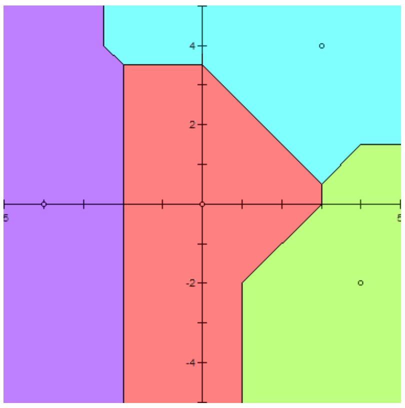
- $N \le 2\,000$

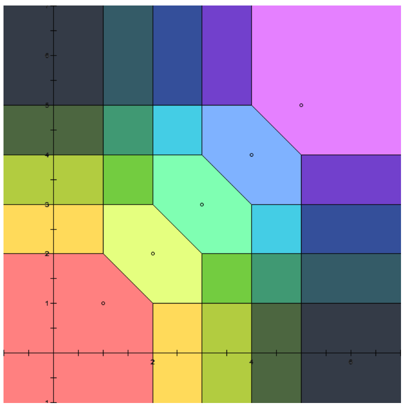
Subtask 3 (50 points)

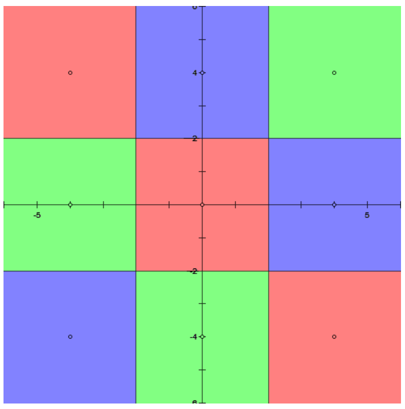
This subtask has no additional constraints.

Examples

standard input	standard output	Notes

4
0 0
-4 0
3 4
4 -2
	4	

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

9
-4 -4
-4 0
-4 4
0 -4
0 0
0 4
4 -4
4 0
4 4
	8	

Note

In example 2, overlapping region is indicated as subtractive mixing of two or more colors. All points with $(x \geq 5 \wedge y \leq 1) \vee (x \leq 1 \wedge y \geq 5)$ is included in all five region, and colored as darkest.

Problem D. A Plus Equals B

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

$A + B$ is a problem used to test one's basic knowledge for competitive programming. Here is yet another boring variation of it.

You have two integers, A and B. You want to make them equal. To do so, you can perform several steps, where each step is one of the following:

- A+=A
- A+=B
- B+=A
- B+=B

Unfortunately, $A + B$ is a hard problem for us, so you are allowed to make at most 5000 steps.

Input

In the first line, two space-separated integers A and B are given. These are the initial values of the variables A and B.

Output

In the first line, print a single integer n ($0 \leq n \leq 5\,000$) denoting the number of steps.

In the next n lines, print one of the following strings to denote your desired operation: "A+=A", "A+=B", "B+=A", or "B+=B".

Any sequence of steps that yields the desired result will be judged correct.

Constraints

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

Subtask 1 (36 points)

This subtask has an additional constraint:

- $A = 1$

Subtask 2 (64 points)

This subtask has no additional constraints.

Examples

standard input	standard output

`2 3`	`4`
	`B+=B`
	`B+=A`
	`A+=A`
	`A+=A`

Problem E. Water Knows The Answers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

Can water think? Can it learn? After reading the book "Water Knows the Answers – the Hidden Messages in Water Crystals", you may believe that water can reflect human emotions and has the power to listen, see, and know the answers to life.

Review of the book "*Water Knows the Answers*", by *Masaru Emoto*.

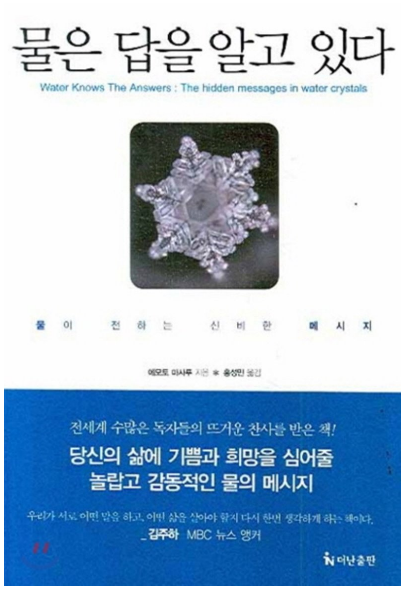

물은 답을 알고 있다
Water Knows The Answers : The hidden messages in water crystals
물이 전하는 신비한 메시지
전세계 수많은 독자들의 뜨거운 찬사를 받은 책!
당신의 삶에 기쁨과 희망을 심어줄
놀랍고 감동적인 물의 메시지
_김주하 MBC 뉴스 앵커

***Figure**: The book "Water Knows the Answers – the Hidden Messages in Water Crystals".*

Donghyun lives in a 2D plane, having x-axis as ground. It rains a lot in the 2D plane, and the rain falls from $y = +\infty$.

Recently, Donghyun read a book called "Water Knows the Answers – the Hidden Messages in Water Crystals", and got impressed by the book. He now believes that water will tell him *the answers to life.*

Donghyun has N rectangular boxes with (possibly) different heights and widths. He is going to rearrange those boxes in order to collect the rainfall. Then, the water will give him the *answers to life.* To achieve this, he is going to place the box in a row. No empty space is allowed between the boxes. He may rotate some of the boxes or not, but he should make its edges parallel to the ground.

The water can flow left or right until it has no space to flow to. Formally, water in certain point can stay in its place if that point is not inside the box and has a box both somewhere to its left and somewhere to its right at the same height.

Donghyun wants to know the maximum area of water he can store. (In the 2D plane, area means volume.) But, he doesn't have water yet, so he doesn't know the answer. Do you know the answer?

Input

Input consists of $N + 1$ lines.

In the first line, the number of boxes N is given.

Next N lines contains the width w_i and height h_i of each boxes, space-separated.

Output

Print a single integer, denoting the maximum area of water Donghyun can store if he arranges the boxes optimally.

Constraints

- $3 \leq N \leq 250,000$
- $1 \leq w_i,\ h_i \leq 10^6$

Subtask 1 (18 points)

This subtask has an additional constraint:

- $N = 3$

Subtask 2 (19 points)

This subtask has an additional constraint:

- $w_i = h_i$

Subtask 3 (34 points)

This subtask has an additional constraint:

- $N \leq 100$

Subtask 4 (14 points)

This subtask has an additional constraint:

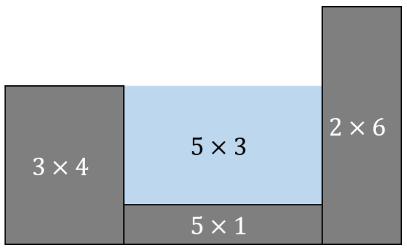
- $N \leq 2\,500$

Subtask 5 (15 points)

This subtask has no additional constraints.

Examples

standard input	standard output	Notes

3
4 3
2 6
5 1
	15	

Problem F. Eat Economically

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

Ho has arrived in a secret place for her secret business trip. She knows her trip will take at most N days or shorter but doesn't know the exact number of days she'll be there. So, **the perfectionist** Ho wants to make the daily meal lists for every possible trip length, 1 day to N days.

There is the only food court that offers exactly $2N$ kinds of menus (by accident) in this secret place. The food court opens only lunch time and dinner time, and oddly, the prices of lunch and dinner for the same menu can be different.

She will eat exactly one menu per lunch and dinner respectively and never eat the same menu for the entire of the trip. She never minds about which kind of menu will be her meal, the only important thing is the entire price of meals must be minimized.

Under these conditions, she can make her meal lists but realizes that writing every $N(N+1)$ menu is hard and tiresome. So, instead of making the meal lists, she calculates the minimized entire price for i lunch menus and i dinner menus where $i = 1$ to N.

You, the big fan of Ho, has a supreme task. Print the N prices she calculated.

Input

The first line contains an integer N.

In the next $2N$ lines, each line contains two integer l, d denoting the prices of the menus when lunch and dinner respectively.

Output

Print N lines. The i-th line ($1 \leq i \leq N$) should contain an integer denoting the minimized entire price for i lunch menus and i dinner menus.

Constraints

- $1 \leq N \leq 250\,000$
- $1 \leq l, d \leq 10^9$

Subtask 1 (16 points)

This subtask has an additional constraint.

- $N \leq 100$

Subtask 2 (33 points)

This subtask has an additional constraint.

- $N \leq 10\,000$

Subtask 3 (51 points)

This subtask has no additional constraints.

Examples

standard input	standard output
1	7
4 9	
5 3	
2	2
1 6	7
2 4	
5 3	
3 1	
4	3
7 5	7
5 7	16
7 4	26
4 2	
2 5	
6 4	
3 2	
1 9	

Problem G. Increasing Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

You are given a permutation of size N. For each i, print the number of indices $j \neq i$, which when removed, decreases the maximum possible length of an increasing subsequence that contains index i.

Input

The first line contains an integer N.

The next line contains N integers $A_1, A_2, \cdots, A_N$.

Output

Print N integers, separated by spaces, denoting the answers for $i = 1, 2, 3, \cdots, N$.

Constraints

- $1 \leq N \leq 250\,000$
- $1 \leq A_i \leq N$
- Every element of A is distinct.

Subtask 1 (22 points)

This subtask has an additional constraint.

- $N \leq 100$

Subtask 2 (17 points)

This subtask has an additional constraint.

- $N \leq 2\,000$

Subtask 3 (61 points)

This subtask has no additional constraints.

Examples

standard input	standard output
1 1	0
6 1 2 3 4 5 6	5 5 5 5 5 5
6 6 5 4 3 2 1	0 0 0 0 0 0
4 2 1 4 3	0 0 0 0
9 1 2 3 6 5 4 7 8 9	5 5 5 6 6 6 5 5 5

Problem H. Jealous Teachers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

There are N teachers and N students in the Korea Science Academy of KAIST (KSA). Each student bought N flowers because tomorrow is a teacher's day in Korea. However, one of the students quit, and now only $N-1$ students remain in the school.

Teachers are very jealous, so they will give an F grade to students when they receive fewer flowers from that student than others. Therefore, every teacher should receive exactly $N-1$ flowers. A student can only give flowers to teachers who have taught him or her, and you know which students have learned from which teachers.

Seunghyun is the student of KSA, and he needs your help in organizing this event.

Input

The first line contains two integers N and M describing the number of teachers and the number of (student, teacher) pairs where the student learned from the teacher.

In the next M lines describe the relations: j-th line contains two integers s_j, t_j indicating that s_j-th student can give flowers to the t_j-th teacher. It is guaranteed that all pairs are different.

Output

If it is impossible to give all teachers the same number of flowers ($N-1$ flowers), print a single number -1 in the first line.

Otherwise, your program should output M lines. In j-th line, there should be a single integer denoting the number of flowers which s_j-th student gave to t_j-th teacher.

If there are multiple possible answers, you can output any of them.

Constraints

- $2 \le N \le 100\,000$
- $1 \le M \le 200\,000$
- $1 \le s_j \le N-1$ $(1 \le j \le N)$
- $1 \le t_j \le N$ $(1 \le i \le N)$

Subtask 1 (32 points)

This subtask has an additional constraint:

- $N \le 100$

Subtask 2 (68 points)

This subtask has no additional constraints.

Examples

standard input	standard output

6 12
1 3
1 4
1 5
2 2
2 4
3 1
3 3
4 1
4 2
4 4
5 4
5 6
	1
	0
	5
	5
	1
	2
	4
	3
	0
	3
	1
	5

6 12
1 2
1 3
1 4
2 2
2 4
3 1
3 3
4 1
4 2
4 4
5 5
5 6
	-1

Problem I. Dijkstra Is Playing At My House

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds

To celebrate your team's victory at ICPC World Finals, Edsger W. Dijkstra (The inventor and namesake of Dijkstra's algorithm) will throw a fabulous party at your house in New York City. The party starts in 4 hours, so he should better start moving.

New York City is modeled as a 2-dimensional plane. Dijkstra is now in coordinate $(s_x,\ s_y)$, and your house is located in coordinate $(e_x,\ e_y)$. Dijkstra should come to your house by only moving in a direction parallel to the coordinate axes (you remember the *Manhattan distance*, right?). Also, there are N skyscraper in an axis-parallel rectangular shape, which you can pass through its boundary, but cannot pass through anywhere strictly inside of it.

You got a phone call from Dijkstra, saying that it's too hard for him to compute the shortest path between his location and your house. Somehow, he is losing his edge. However, that's not bad news, because it's a chance for you to be cool in front of the great Dijkstra. Can you?

It is guaranteed that all x coordinates are distinct and all y coordinates are distinct. It is also guaranteed that no pair of rectangles overlap. It is also guaranteed that your house and Dijkstra's starting location are not inside of any rectangles.

Input

The first line contains five space-separated integers N, s_x, s_y, e_x, e_y.

The i-th line of next N lines contain four space-separated integers a_i, b_i, c_i, d_i. This indicates that i-th skyscraper is a rectangle with its four corners located in $(a_i,\ b_i)$, $(a_i,\ d_i)$, $(c_i,\ b_i)$, $(c_i,\ d_i)$.

Output

Print the length of the shortest path between Dijkstra's location and your house, using the Manhattan metric.

Constraints

- $1 \le N \le 250\,000$
- $0 \le s_x,\ s_y,\ e_x,\ e_y \le 10^8$
- $0 \le a_i < c_i \le 10^8$ $(1 \le i \le n)$
- $0 \le b_i < d_i \le 10^8$ $(1 \le i \le n)$
- Let $X = \{s_x,\ e_x,\ a_1,\ a_2,\ \cdots,\ a_N,\ c_1,\ c_2,\ \cdots,\ c_N\}$, all elements in X are distinct.
- Let $Y = \{s_y,\ e_y,\ b_1,\ b_2,\ \cdots,\ b_N,\ d_1,\ d_2,\ \cdots,\ d_N\}$, all elements in Y are distinct.
- No pair of rectangles share a common point.
- Dijkstra's location and your house's location are not in any of the rectangles.

Subtask 1 (31 points)

This subtask has an additional constraint.

- $N \le 500$

Subtask 2 (24 points)

This subtask has an additional constraint.

- $N \leq 5\,000$

Subtask 3 (45 points)

This subtask has no additional constraints.

Examples

standard input	standard output

3 2 14 5 1
4 6 6 10
0 7 3 9
1 2 8 5
	20

1 0 500 100 503 1 0 99 1000	1097

2 2 8 10 3
3 6 6 10
7 1 8 7
	15