Judging Notes

1. Very few clarifications are issued during the contest, and they are only for ambiguities. Please read the problem statement and examine the sample test cases carefully before submitting any request for clarifications.
2. The following guidelines apply to handling input/output in programs:
   - All input comes from standard input.
   - All output goes to standard output.
   - Unless the problem explicitly states otherwise, the input for a problem consists of a single test case. If the input contains multiple test cases, then the problem will also state an upper bound on the number of test cases.
   - Your program may be run on multiple input files. Note that this means that if your program has more than one error (say, Time Limit Exceeded and Wrong Answer), then you can get either error as judgment.
   - Output formatting should follow the specification and sample output in the problem statement.
3. For problems with floating point output, the judging system will accept a range of answers as correct as long as they satisfy the constraints described in the problem statement. These constraints will be specified as an absolute and/or relative tolerance, which will be given.
4. There is no such thing as "Presentation Error" or "Format Error." If you misspell the word "impossible," for example, and the problem requires that word as output, then your submission will be judged as "Wrong Answer."
5. Unless a problem specifically indicates that uppercase or lowercase letters are important, then either will be accepted. For example, "Yes" or "yes" would be treated the same, but "yse" is Wrong Answer.
6. You should follow the sample output format, but extra whitespace within reason is acceptable. For example, if you print out a gigabyte of blanks, then the judging system will treat that as Wrong Answer; however an extra blank at the end of a line or an extra blank line between test cases is acceptable.
7. During the contest, input size constraints on test cases will be given as part of the problem statements.
8. If you submit a solution that has a Compile Error, then you will be notified of it (just as any other error). However, Compile Errors do not count toward penalty time.

Problem A : Micromasters

The Department of Computer Engineering at Sharif University of Technology has recently initiated a professional education program known as Micromasters. This program offers a set of courses designed to empower students with specialized knowledge and skills in various domains of computer science and engineering. As an incentive to promote the program, the department has introduced a referral system wherein individuals who refer other students to the Micromasters program receive a 10% discount for each referred student on their own course registrations.

Mina is a talented student who is passionate about spreading the benefits of the Micromasters program. With each referral, Mina's list of discounts grows, and now the following question arises: given the number of students who are referred by Mina, how many courses can she enroll in for free?

Input

The input consists of a single line containing a single integer n ($0 \leqslant n \leqslant 1000$), which represents the number of students that Mina has referred.

Output

Print a single line, containing the number of courses Mina can enroll in for free using the discounts.

Example

Standard Input	Standard Output
5	0

Standard Input	Standard Output
18	1

Problem B : Hezardastan's Annual Report

Hezardastan, a giant among Iranian IT holding groups, houses several innovative companies such as Cafebazaar, Divar, and Balad. The annual report of the holding consists of n chapters, each dedicated to a company under Hezardastan's umbrella. The chapters in the report vary in length and occupy a certain number of pages. We want to compile all n chapters into a PDF document that will be printed double-sided on A4 paper sheets. However, for aesthetic reasons, we want to avoid having pages from two different chapters printed on the same paper sheet. To ensure each chapter begins on a fresh, odd-numbered page, we plan to strategically insert an extra blank page after each chapter that has an odd number of pages. Now, we need to know the minimum number of A4 paper sheets needed to print the entire holding company report?

Input

The input consists of two lines. The first line contains a single integer n ($1 \leqslant n \leqslant 100$), the number of chapters in the report. The second line contains n space-separated integers, denoting the number of pages in each chapter. All numbers in the input are positive integers and are at most 100.

Output

The output should consist of a single line containing the total number of A4 paper sheets needed to print the entire annual report.

Example

Standard Input	Standard Output
5 1 1 2 1 2	5

Standard Input	Standard Output
8 1 2 3 2 2 5 4 2	12

Problem C : Moderation in all things

Initially, we have an array of length 1 containing only the number 0. All natural numbers are listed in ascending order in the "reservation list" (the first number in the list is 1). The array undergoes q operations. The i^{th} operation, is one of the following:

- `Insert(p, x)`: Insert the first x numbers from the reservation list after the number p in the array, in ascending order. These numbers are removed from the reservation list.
- `Remove(p, x)`: Remove the next x numbers after number p in the array. These numbers are not returned to the reservation list.

You are given information about q operations, and you are asked to determine the number written in the middle of the array after each operation. If the length of the array after the i^{th} operation is n, you should find the $\lceil \frac{n}{2} \rceil^{\text{th}}$ element of the array. Note that the indexing of the array starts from 1.

Input

The first line contains an integer q ($1 \leqslant q \leqslant 5 \cdot 10^5$), which represents the number of operations. Each of the next q lines contains two integers: p_i ($1 \leqslant p_i \leqslant 2 \cdot 10^9$), and k_i ($1 \leqslant |k_i| \leqslant 2 \cdot 10^9$).

If $k_i = +x$, operation `Insert(`p_i`, x)` is executed. If $k_i = -x$, operation `Remove(`p_i`, x)` is executed. It is guaranteed that all operations are valid, and no impossible operation is performed on the array. Additionally, at most $2 \cdot 10^9$ numbers are moved from the reservation list into the array.

Output

Output q lines. In the i^{th} line, print the middle element of the array after performing the i^{th} operation.

Example

Standard Input	Standard Output
10 0 3 0 2 5 -2 4 1 0 -2 5 2 7 3 3 2 10 5 12 20	1 5 4 6 5 7 9 10 16 22

Problem D : Cup of Tea

Abolf lives in Aboland, a country consisting of n cities and $n-1$ two-way roads. In Aboland, one can travel from any city to any other city using these roads. Aboland's cities are numbered from 1 to n.

Abolbucks is a multinational chain of teahouses which serves the best tea in the world. When Abolf enters a city with an Abolbucks branch, he drinks a cup of tea and instantly reaches k units of happiness. However, each time Abolf travels through the i^{th} road, he must pay c_i coins as toll which causes him to lose c_i units of happiness.

Abolf currently resides in city 1 and wants to plan his summer trip. If at any point during his trip Abolf's happiness drops below zero, he would stops his trip immediately. For each city t (for $2 \leqslant t \leqslant n$), Abolf wants to know what is the minimum amount of coins he should pay to reach city t while making sure that his happiness remains non-negative at all time, including at the destination.

He has asked you to find this amount for each city except for his home city. Note that each destination should be considered separately. Also, he may visit a city multiple times during his trip.

Input

The first line of input contains two integers n and k ($2 \leqslant n \leqslant 3 \cdot 10^5$, $1 \leqslant k \leqslant 10^9$), the number of cities in Aboland and Abolf's happiness after he drinks a cup of tea, respectively. Each of the next $n-1$ lines contains three space-separated integers v_i, u_i, and c_i ($1 \leqslant v_i, u_i \leqslant n, 1 \leqslant c_i \leqslant 10^9$, $u_i \neq v_i$) indicating that the i^{th} road connects city u_i and city v_i, and Abolf should pay c_i coins each time he travels through this road. The last line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \leqslant a_i \leqslant 1$). If $a_i = 1$, there is an Abolbucks branch in city i. It is guaranteed that $a_1 = 1$.

Output

In the only line of the output, you should print $n-1$ integers. The i^{th} number should be the minimum amount of coins it takes for Abolf to reach city $i+1$ from city 1. If there is no way to reach city $i+1$ such that Abolf's happiness remains non-negative at all time, print -1 for that city.

Example

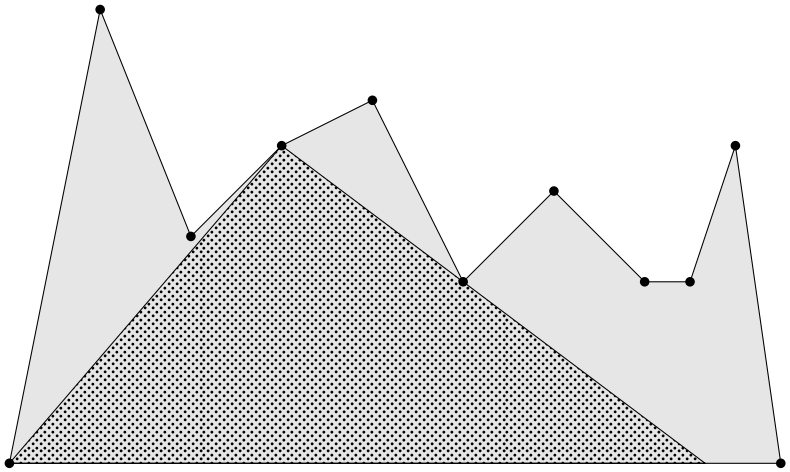
Standard Input	Standard Output
6 3 1 2 4 1 3 3 1 4 2 4 5 1 4 6 2 1 1 0 0 1 0	-1 3 2 3 6

Problem E : Largest Triangle

A "terrain" is an x-monotone polygon defined by the points $p_1, \ldots, p_n$ where each point p_i has coordinates (x_i, y_i), and the following three conditions hold:

- $y_1 = y_n = 0$
- $y_i > 0$ for $1 < i < n$
- $x_i < x_{i+1}$ for $1 \leqslant i < n$

Given a terrain defined by the points $p_1, \ldots, p_n$, find the largest triangle that fits entirely within the terrain, and one of its three vertices is positioned at one of the terrain points p_2 through p_{n-1}.



Input

The first line of input contains an integer n, representing the number of points in the terrain ($3 \leqslant n \leqslant 10^5$). The i^{th} line in the following n lines consists of two space-separated integers x_i and y_i, representing the point p_i of the terrain ($0 \leqslant x_i, y_i \leqslant 10^9$).

Output

Print the area of the largest triangle contained within the terrain. Your output will be considered correct if its absolute or relative error is at most 10^{-6}.

Example

Standard Input	Standard Output
11 0 0 2 10 4 5 6 7 8 8 10 4 12 6 14 4 15 4 16 7 17 0	53.666667

Problem F : Micromasters Certificates

The Department of Computer Engineering has provided several micromasters, each containing a curriculum. If a student successfully completes all the courses of a micromaster, he will receive the certificate of that micromaster. A course may be included in the curriculum of several micromasters. Soroush, who only thinks about getting a certificate and doesn't care about the type of certificate, wants to get 3 micromasters certificates by taking the minimum possible number of courses. The micromasters curriculums are posted on the bulletin board. Help Soroush reach his goal according to the micromasters curriculums.

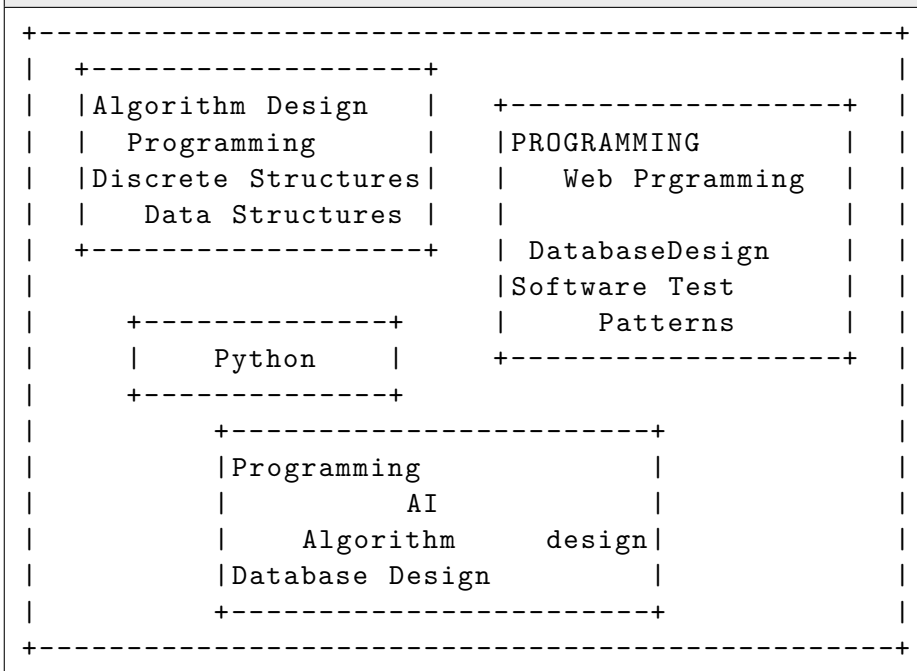
Input

The input represents a bulletin board. The board consists of at most 400 rows and 400 columns. Each micromasters curriculum is encapsulated in a rectangular box. The boundaries of the bulletin board and the curriculum boxes are represented by characters "+", "-", and "|" for corners, horizontal sides, and vertical sides, respectively. The curriculum boxes are disjoint (with no characters in common) and each has its own boundary. Each line inside a curriculum box contains at most one course name. Course names consist of alphanumeric and space characters. Course names are not case-sensitive, and spaces do not matter in them. For example, "`General  math1`" and "`generalMath 1`" are the same. There are at most 50 curriculum boxes and each box contains at most 30 courses. It is guaranteed that there are at least 3 boxes on the board and there is at least 1 course in each box.

Output

Print a single line containing the minimum number of courses that should be taken by Soroush to get at least 3 certificates.

Example

Standard Input	Standard Output
	8

Problem G : Jackson House

Jackson, after witnessing the advancements in the world of technology, decided to sell his small cozy house and enroll in the programming-and-algorithm micromaster. He came across an interesting algorithm that he needed to analyze and solve the problem related to it, in order to pass the exam at this stage of the course. The pseudocode of this algorithm is as follows:

```
input: a permutation π = ⟨π_1, π_2, ..., π_n⟩ of numbers {1, 2, ..., n}
while π is changing during this iteration:
    for i := n downto 2:
        if π_i < π_⌊i/2⌋:
            swap(π_i, π_⌊i/2⌋)
```

He wants to know for how many permutations π of length n from the possible $n!$ ones, the final permutation will be sorted after running this algorithm.

Input

The first line contains an integer t ($1 \leqslant t \leqslant 100$), the number of test cases.

Each of the next t lines contains an integer n_i ($2 \leqslant n_i \leqslant 10^9$), representing the length of the permutation for the i^{th} test case.

Output

Output t lines. On the i^{th} line, print the number of permutations of length n_i which will be sorted after running the provided algorithm on it. Since the output could be very large, output the result modulo $10^9 + 7$.

Example

Standard Input	Standard Output
4 3 5 10 20	4 16 1728 23887872

Problem H : Star Wars

Amirreza is playing the Star Wars game. This game is played on an $n \times m$ board where each cell is either empty ('.'), contains a white piece ('W') or a black piece ('B'). At start of the game, Amirreza should choose exactly one white piece to play with. Afterwards he can move this piece multiple times to knock out as many black pieces as possible. Suppose the white piece is currently in cell (i, j) of the board; In one move, this piece can go up-left $(i-1, j-1)$, up $(i-1, j)$ or up-right $(i-1, j+1)$, provided that cell exists on the board and it does not contains another white piece. If the cell contains a black piece, it will be knocked out. Help Amireza figure out the maximum number of black pieces he can knock out.

Input

The first line contains two integers n and m $(1 \leqslant n, m \leqslant 50)$, the number of rows and columns in the board, respectively. This is followed by n lines, each containing m characters. The j^{th} character of the $(i+1)^{\text{th}}$ line represents cell (i, j). Each character is 'W', 'B', or '.', denoting a white piece, a black piece, or an empty cell, respectively.

Output

Print a single integer, the maximum number of black pieces Amirreza can knock out.

Example

Standard Input	Standard Output
8 10 .W...BB... W..B.WB... .B.WB...W. .B..B..... ..W...BB.. B.B..B.W.W .WB.W...B. ..W..BW.B.	5

Problem I: Pistons

Maryam, a famous mathematician, recently has bought an old vintage car. This car uses a combustion engine to generate the power needed to move the car. Inside the engine, there are n cylinders of length m and inside each cylinder, there is a piston constantly moving up and down. All pistons move independently and at the same speed. At any given time, the position of a piston inside a cylinder can be shown with an integer from 0 to m, which also describes the area of the cylinder occupied by the piston. A piston instantly changes its direction when it reaches the top (position m) or bottom (position 0) of its cylinder.

Maryam managed to determine the position and direction of all the pistons at a specific time. Now she is curious about the maximum total area occupied by all the pistons. Help Maryam find out this value.

Input

The first line of input contains two integers n and m ($1 \leqslant n \leqslant 10^5, 1 \leqslant m \leqslant 10^6$), describing the number of pistons and the length of cylinders, respectively. Each of the next n lines describe the position and direction of a single piston. The $(i+1)^{\text{th}}$ line of the input contains an integer p_i ($0 \leqslant p_i \leqslant m$), and a character d_i ($d_i \in \{\texttt{U}, \texttt{D}\}$), the initial position of the i^{th} piston and its direction (Up or Down), respectively.

Output

Print a single integer, the maximum total area occupied by all the pistons.

Example

Standard Input	Standard Output
2 5 2 U 5 D	7

Standard Input	Standard Output
4 6 0 U 0 D 6 U 3 U	15

Problem J : Cafebazaar's Applications

It's the end of the year, and Cafebazaar has released a list, containing the number of users of each of its n applications. Now, each application is eager to showcase its success through an advertisement image, which highlights a continuous subset of the application list containing the application itself. Also, for the image to be credible, it should contain at least k applications, including itself.

For each application in this list, we need to determine the minimum possible rank this application can achieve within any valid subset, according to the number of users. The rank of an application within a subset is defined by the number of applications in that subset that have more users than it, plus one.

دیوار	15000000
بازاریاب	10000000
phanous	30000000
کارنامه	20000000
	200000
	70000000
	100000000

Input

The first line of input consists of two integers n and k ($1 \leqslant k \leqslant n \leqslant 10^5$), where n represents the total number of applications and k represents the minimum number of applications in an advertisement image. The following n lines contain information about each application: the i^{th} line contain c_i, representing the number of users for the i^{th} application ($1 \leqslant c_i \leqslant 10^8$).

Output

In the only line of output print n space-separated integers. The i^{th} integer should be the minimum rank that i^{th} application can achieve within an advertisement image.

Example

Standard Input	Standard Output
7 3 15000000 10000000 30000000 20000000 200000 70000000 100000000	2 3 1 2 3 1 1

Standard Input	Standard Output
3 2 10 10 10	1 1 1

Problem K : Monster Warehouse

Mike and Sally work in the warehouse of Monster Inc. as storekeepers. Their daily tasks are to process incoming requests and update the inventory of the Warehouse. Requests only include buying, selling, unpacking, and packing containers. The warehouse includes goods and containers and has unlimited space. A container may contain goods or other containers as sub-containers.

The exact format of the requests is given below.

- `BUY <CONTAINER_DESCRIPTION>`
- `SELL <CONTAINER_ID>`
- `UNPACK <CONTAINER_ID>`
- `PACK <CONTAINER_DESCRIPTION>`

Each container which is not inside any other container is uniquely identified by a positive integer ID. Assigning IDs to containers is done sequentially and started from 1. An ID is valid if and only if its container exists in the warehouse, otherwise it is invalid.

A container description is enclosed in parentheses and lists the contents, which can be either goods or sub-containers. A good is identified exclusively by its name, which consists of non-case-sensitive English letters. Multiple units of a good may be available. To denote quantities, place a positive integer '`N`' before or after the good name (separated by one whitespace), where $N < 100$ is the number of the good. For example, `((tomato, potato), 4 celery, (wood, (silk 3, banana 2)))` describes a container with four units of celery and two sub-containers.

The description of each request is as follows:

- `BUY`: A new container is transferred into the warehouse and an ID is assigned to it.
- `SELL`: An existing container with the given ID is ship out and its ID becomes invalid.
- `UNPACK`: All goods and sub-containers are extracted from the container and added to the warehouse. Moreover, the sub-containers become new containers and get their own ID. The assignment of IDs to the new containers is based on the order of their appearance in the container description (from left to right). For instance, considering the following two lines as the first requests, results in adding one unit of celery and adding three containers with IDs 2, 3, and 4 to the warehouse and ID 1 becomes invalid.

```
BUY (celery, (Banana), (Celery), (celery))
UNPACK 1
```

- `PACK`: Goods specified in a `PACK` request are grouped into a new container, which is then assigned the next available ID.

Mike and Sulley process the requests in the order they are received. Any request with an invalid container ID must be discarded. Moreover, for `PACK` request they need to check if there exists enough units of each good in the warehouse.

Roz, the agent of Monster Inc. has told Mike once "I'm watching you, Wazowski. Always watching. Always.", She rolled her desk into their office and asked for requests and reports. She is looking for every detail. She is reviewing each request and might ask a few questions. Her questions might be each of the following types:

- `? COUNT <good>`: How many units of the given good exist outside of containers?
- `? CONTAINS <good>`: How many containers with ID have the given good, i.e. the good is in the container or there is a recursive sub-container which contains that good.
- `? MIN <good>`: At least how many containers should be unpacked to reach one unit of the good. If it is impossible, the answer should be `-1`.

Mike and Sully are expected to answer these queries using just one integer.

Before helping Mike and Sully, read samples carefully.

Input

The input consists of n requests or queries from Roz while she is reviewing ($1 \leqslant n \leqslant 5000$); each appears in a separated line. The name of each good is limited to 100 characters. Each container description might have at most 5000 characters and the input size is less than 10^6 characters.

Output

Each line of the report is associated with a request or Roz's questions. After each `BUY`, `SELL`, `PACK`, `UNPACK` request, you should print `OK`, if the request is not discarded. Otherwise, print `DISCARD`. If the request is `UNPACK`, after printing `OK`, you should print the number of containers added to the warehouse (read samples for more details). For each Roz's query, print just one integer in a line.

Example

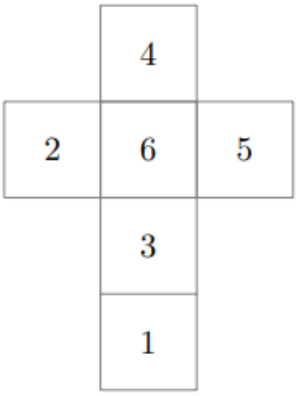
Standard Input	Standard Output

`BUY (10 apple, 10 carrot, 10 banana, ())`
`UNPACK 1`
`UNPACK 2`
`PACK (3 apple, (5 cArrot, 2 banana))`
`PACK (3 apple, (5 carrOt, 1 banana))`
`PACK (5 apple, (3 banana))`
`? MIN apple`
`? MIN CaRRoT`
`? CONTAINS apple`
	`OK`
	`OK, 1 container added.`
	`OK, No containers added.`
	`OK`
	`OK`
	`DISCARD`
	`0`
	`2`
	`2`

Standard Input	Standard Output
BUY (apple, (banana, 3 carrot)) BUY ((banana, apple), (banana, carrot)) ? MIN apple UNPACK 1 ? COUNT apple ? COUNT carrot ? CONTAINS banana	OK OK 1 OK, 1 container added. 1 0 2

Standard Input	Standard Output
BUY ((duck 2, carrot), 1 celery) ? MIN duck ? MIN carrot ? MIN celery ? MIN test SELL 10 PACK (celery) UNPACK 1 UNPACK 1 PACK (celery) ? COUNT celery ? COUNT carrot ? CONTAINS celery ? CONTAINS carrot BUY ((duck 8, carrot), ((7 duck), celery)) UNPACK 4 UNPACK 5 UNPACK 6 ? COUNT DUCK ? MIN duck	OK 2 2 1 -1 DISCARD DISCARD OK, 1 container added. DISCARD OK 0 0 1 1 OK OK, 2 containers added. OK, No containers added. OK, 1 container added. 8 0

Problem L : Rolling-Dice Puzzle



Sarina and her brother, Soroush, are playing the rolling-dice game. The game is played on an $n \times m$ board. Initially, Soroush places a standard dice in one of the cells. It is place in a way that the number 6 is on the upper face, the number 4 is on the north face, and the number 2 is on the west face. In a standard dice, 6 is on the opposite side of 1, 2 is on the opposite side of 5, and 3 is on the opposite side of 4. Additionally, he selects some of the cells and writes arbitrary integers numbers from 1 to 6 in them.

After that, Sarina have to move the dice on the board by rolling it multiple times. The act of rolling is defined as follows: Suppose two adjacent cells A and B share an edge e and the dice is on the cell A; The dice can be rolled around its edge incident to e and moved from A to B. For example, consider the starting position of the dice. If the dice is rolled around the east, west, north, and south edges, the number appearing on the top face after rolling will be 2, 5, 3, and 4, respectively.

Whenever Sarina moves the dice to a a cell with a number in it in such a way that the number on the upper face of the dice matches the number in that cell, she gets a point. Note that Sarina can get a point from each cell at most one. The game is not that simple! There are obstacles in some of the cells and it is not possible to move the dice to the cells with an obstacle in it. Your task is to find out the maximum points that Sarina can get.

Input

The first line of input contains two integers n and m ($1 \leqslant n, m \leqslant 100$), indicating the number of rows and columns of the board, respectively. Each of the next n lines contain m characters, describing the board. Empty cells are represented by "." and obstacles are represented by "x". The starting position of the dice is represented by "s" and the selected cells are represented by the integers written in them (from 1 to 6). It is guaranteed that there is only one "s" in the input.

Output

Output a line containing the maximum points Sarina can get.

Example

Standard Input	Standard Output
3 4 .23s 4.2x xx.1	5

Standard Input	Standard Output
2 2 4s 22	1

Problem M : Colorful Intervals

The *Museum of Contemporary Art* is holding a painting gallery focused on modern art, especially *Monochromatic style* paintings, which use only a single color. The gallery displays n paintings arranged in a line.

The ICPC wants to bring students on an excursion to the gallery to spark their interest in art. However, the students are programmers, and everyone knows programmers only care about the colors of these modern paintings. They are also somewhat impatient. To keep their attention and to ensure they see every color without overwhelming them, the organizer decided to show them exactly two intervals of painting. This approach balances their short attention span and ensures all colors are represented. The task is to find two intervals of paintings such that each color appears at least once in at least one of the intervals, and the total number of paintings the students need to see is minimized.

Input

The input consists of a single line containing a non-negative integer n ($2 \leqslant n \leqslant 2000$), indicating the number of paintings. This is followed by n lines, each containing a string representing the color of a painting. Each color is represented by a non-empty lowercase string with a length of less than 20. It is guaranteed that there are at least 2 and at most 50 different colors in the input.

Output

In the output, print the minimum number of paintings the ICPC students need to see, which is the sum of the lengths of the two intervals.

Example

Standard Input	Standard Output
5 blue red blue black red	3

Standard Input	Standard Output
8 peachfuzz livingcoral livingcoral teal teal livingcoral livingcoral coral	5