Singapore Informatics League 2025

Task Statements

December 2025

Contents

Level 1

Task 1: `Broken Key`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133)

Statement

Bob wants to enter a positive integer n on his calculator. However, the digit key k on his calculator is broken! To work around this, Bob plans to express n as a sum of one or more integers that can be typed using only the working digit keys (and the $+$ key).

Help Bob determine the minimum number of such integers he needs to sum in order to create n.

You may assume that n can always be formed using only the functioning digit keys and the $+$ sign.

Input Format

The first line of input contains two space-separated integers n and k.

Output Format

Output the minimum number of integers Bob needs to type to create n.

Input Constraints

Subtask 1: $n = 67$ and $k = 8$

Subtask 2: $n = 199$ and $k = 1$

Subtask 3: $n = 20490$ and $k = 0$

Sample Testcase

Input	Output
19 9	2

Sample Testcase Explanation

In the first sample test case, the digit key 9 is broken. Bob can express 19 as $11 + 8$.

Input File Link

Link to the Input Files

Task 2: `Cats and Parking`

Authored by: Ernest Kiew (Shor the Duck)

Prepared by: Chua Wee Chong (sheep) and Ernest Kiew (Shor the Duck)

Statement

There are n parking spots in a row, numbered from 1 to n. Some spots have cats sitting in them. You are given a string s of length n that shows which spots are occupied: $\texttt{o}$ means there is a cat, and $\texttt{x}$ means the spot is empty.

Each day, every cat moves one spot to the right. A cat on spot $i < n$ moves to spot $i + 1$, and a cat on spot n moves to spot 1.

What is the largest number of consecutive empty spots that ever appear on any day?

Input Format

The first line of input contains one integer n, the number of parking spots.

The second line of input contains a string s of length n, describing which parking spots are occupied on the first day.

Output Format

Output the largest number of consecutive empty spots that ever appear on any day.

Input Constraints

Subtask 1: $n = 15$, all parking spots are empty.

Subtask 2: $n = 15$

Subtask 3: $n = 26$

Sample Testcase

Input	Output
6 oxoxxo	2

Sample Testcase Explanation

In the first sample test case, the configurations of the cats in the parking spots across the first six days are: oxoxxo, ooxoxx, xooxox, xxooxo, oxxoox, and xoxxoo.

On the seventh day, the configuration loops back to the initial configuration. The largest number of consecutive x's (empty spots) in a row is 2.

Input File Link

Link to the Input Files

Task 3: `Distribution`

Authored by: Chua Wee Chong (sheep)

Prepared by: Brian Lee (penguin133) and Chua Wee Chong (sheep)

Statement

There are n people who wish to distribute a total of t apples among themselves. However, no two people can get the same amount of apples. What is the largest possible number of apples that the person who gets the least can have?

You are guaranteed that at least one distribution exists for the given values of n and t.

Input Format

The first line of input contains two space-separated integers n and t.

Output Format

Output the largest possible number of apples that the person who gets the least can have.

Input Constraints

Subtask 1: $n = 2$ and $t = 100$

Subtask 2: $n = 10$ and $t = 55$

Subtask 3: $n = 20$ and $t = 300$

Sample Testcase

Input	Output
3 10	2

Sample Testcase Explanation

In the first sample test case, there are $n = 3$ people and $t = 10$ apples. One possible optimal distribution is 2, 3, and 5 apples. The output is 2.

Input File Link

Link to the Input Files

Task 4: `Duck Fight`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Ryan Goh (rgca)

Statement

Shor the Duck and oolimry the Duck are playing a card game. There are n cards laid out in a row on the table, labelled from 1 to n from left to right. The i-th card from the left has power $a[i]$.

Shor chooses some of the cards first, and oolimry takes all the remaining cards. Shor wins if the total power of the cards he picks is strictly greater than the total power of the cards left for oolimry.

Shor wants to win while picking as few cards as possible. What is the minimum number of cards he needs to choose to win?

Input Format

The first line of input contains one integer n, the number of cards on the table.

The second line of input contains n space-separated integers $a[1]$, $a[2]$, $\ldots$, $a[n]$, representing the powers of the cards.

Output Format

Output the minimum number of cards Shor needs to choose to win.

Input Constraints

For all subtasks, $1 \leq a[i] \leq 10$ for all $1 \leq i \leq n$.

Subtask 1: $n = 12$ and $a[1] = a[2] = \cdots = a[n]$

Subtask 2: $n = 5$

Subtask 3: $n = 12$

Sample Testcase

Input	Output
4 5 2 1 3	2

Sample Testcase Explanation

In the first sample test case, Shor needs to take the card with power 5. Then, he can choose any of the remaining cards and his total will be strictly more than the power of oolimry's cards. This yields the minimum number of cards of 2.

Do note that if Shor only takes the card with power 5, oolimry will have $2 + 1 + 3 = 6$ power, which would cause Shor to lose.

Input File Link

Link to the Input Files

Task 5: `Bench`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133)

Statement

There is a bench with n seats, numbered 1 to n from left to right. The seats are evenly spaced 1 unit apart, so the distance between two people sitting in seats x and y is $|x - y|$ units.

There are m people who wish to sit on the bench. No two people can sit in the same seat. They also wish to be as far apart from each other as possible.

Let d be the smallest distance (in units) between any two people in the seating arrangement. What is the largest possible value of d?

Input Format

The first line of input contains two space-separated integers n and m.

Output Format

Output the largest possible value of d.

Input Constraints

Subtask 1: $n = 4278$ and $m = 2$

Subtask 2: $n = 8$ and $m = 3$

Subtask 3: $n = 9996$ and $m = 35$

Sample Testcase

Input	Output
6 3	2

Sample Testcase Explanation

In the first sample test case, there are $n = 6$ seats and $m = 3$ people. It can be proven that the largest possible of d is 2. This can be achieved, for instance, by using the seats 1, 3, and 6.

Input File Link

Link to the Input Files

Task 6: `Schedule`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133) and Chua Wee Chong (sheep)

Statement

Alice is scheduled to work for n days. Her work schedule is described by a string s of length n. On day i ($1 \leq i \leq n$), the character $s[i]$ describes the status on that day:

- 'W' means that she is working on day i
- 'R' means that she is resting on day i
- 'X' undecided; she may choose to work or rest on this day

Alice wishes to replace every 'X' in s with either 'W' or 'R' such that she maximises the number of working days. However, she also wishes to ensure that she never works for more than k consecutive days.

Under this condition, what is the largest number of days Alice can be working? It is guaranteed that there is at least one valid replacement.

Input Format

The first line of input contains two space-separated integers n and k, the number of days Alice has been scheduled to work for and the maximum number of days in a row she can work for.

The second line of input contains a string s of length n, the string of characters representing her schedule.

Output Format

Output the largest number of days Alice can work in the n days without working more than k days in a row.

Input Constraints

For all subtasks, $s[i] \in$ 'W', 'R', 'X' for all $1 \leq i \leq n$.

Subtask 1: $n = 10$ and $k = 3$

Subtask 2: $n = 30$, $k = 6$ and $s[i] =$ `X` for all $1 \leq i \leq n$

Subtask 3: $n = 30$ and $k = 4$

Sample Testcase

Input	Output
`8 2` `WXXWWXRX`	`5`

Sample Testcase Explanation

In the first sample test case, a schedule that achieves 5 working days is "$WWRWWRRW$". This is a valid schedule because Alice will only work at most 2 days consecutively.

An example of an invalid schedule is "$WRWWWRRR$". This is because Alice will work 3 days consecutively from days 3 to 5, which exceeds the limit of $k = 2$ days.

Input File Link

Link to the Input Files

Task 7: `Typing Test`

Authored by: Chua Wee Chong (sheep)

Prepared by: Chua Wee Chong (sheep)

Statement

Brian has a string s of length n. This string is made up only of the characters '`a`' and '`A`'. He wishes to type this string one character at a time using only the '`a`' key and the Caps Lock key. Initially, the Caps Lock is off.

Before he begins typing, Brian is allowed to change at most one character of s, from '`a`' to '`A`' or '`A`' to '`a`'. He can also leave the string unchanged.

What is the minimum number of Caps Lock presses required for Brian to type the string s, after changing at most one character?

Input Format

The first line of input contains one integer n, the length of the string s.

The second line of input contains a string s of length n, the string to be typed.

Output Format

Output the the minimum number of Caps Lock presses required for Brian to type the string s, after changing at most one character.

Input Constraints

For all subtasks, $s[i] \in$ '`a`', '`A`' for all $1 \leq i \leq n$

Subtask 1: $n = 9$, there is exactly one '`a`' in s

Subtask 2: $n = 20$

Subtask 3: $n = 20$

Sample Testcase

Input	Output
4 aAAa	1

Sample Testcase Explanation

In the first sample test case, Brian can change the last character of s. Now, $s =$ "aAAA". To type this string, he will need to press the Caps Lock key only once (before typing the second character).

In the second sample test case, Brian can choose not to change any characters of s. Since the Caps Lock is off initially, he will need to press Caps Lock once to type $s =$ "AA".

Input File Link

Link to the Input Files

Task 8: `Ducks, Rabbits and Weirdos`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Brian Lee (penguin133) and Chua Wee Chong (sheep)

Statement

On a farm there are three kinds of animals: ducks, rabbits, and weirdos. Every animal has exactly 1 head.

Ducks have 2 legs each, and rabbits have 4 legs each. Weirdos are a little special. The number of legs each weirdo has is equal to the number of weirdos on the farm. In other words, if there are w weirdos, then each weirdo would have w legs.

You count a total of h heads and l legs on the farm. What is the smallest possible value of w (the number of weirdos) that can achieve the observed values of h and l?

It is guaranteed that there is at least one valid allocation of animals matching the given h and l.

Input Format

The first line of input contains two space-separated integers h and l, the total number of heads and legs that you count on the farm respectively.

Output Format

Output the smallest possible value of w that is consistent with the observed totals of h and l.

Input Constraints

Subtask 1: $h = 8$ and $l = 40$

Subtask 2: $h = 20$ and $l = 119$

Subtask 3: $h = 407$ and $l = 2819$

Sample Testcase

Input	Output
4 11	1

Sample Testcase Explanation

In the first sample test case, one possibility that minimises the number of weirdos is the following: there is 1 duck, 2 rabbits and $w = 1$ weirdo on the farm. This would mean that the weirdo has $w = 1$ leg.

This gives us a total of $1 + 2 + 1 = 4$ heads and $1 \times 2 + 2 \times 4 + 1 \times 1 = 11$ legs, which is consistent with the observed values of $h = 4$ and $l = 11$. Hence, output $w = 1$.

Input File Link

Link to the Input Files

Level 2

Task 9: AANDNANDT

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Kang Yiming (kym2006)

Statement

Shor the duck is trying to spell a word. He does so by reading the word character by character and saying the word `AND` between consecutive characters. For example, if the word is ANT, he says "A AND N AND T".

A speech-to-text program records what Shor says but removes all spaces, so the recorded string becomes `AANDNANDT`.

You are given the recorded string s, and would like to know: how many characters `A` does the original word contain?

It is guaranteed that s is always a valid recording produced in this way from some non-empty original word.

Input Format

The first line of input contains a string s, the recorded string.

Output Format

Output the number of characters `A` in the original word.

Input Constraints

For all subtasks:

- s consists only of uppercase characters.
- s has the form $l_1\texttt{AND}l_2\texttt{AND}\ldots\texttt{AND}l_k$, where l_1, l_2, $\ldots$, l_k are single uppercase characters and $k \geq 1$.

Subtask 1: $|s| = 29$

Subtask 2: $|s| = 233$

Subtask 3: $|s| = 199997$

Sample Testcase

Input	Output
AANDNANDT	1

Input File Link

Link to the Input Files

Task 10: `90RP Manifestation`

Authored by: Ernest Kiew (Shor the Duck)

Prepared by: Ryan Goh (rgca)

Statement

Oh no! Shor the Duck is in desperate need of 90RP. So he decides to create his own `90`.

He starts with a string s of length n, consisting only of numeric digits. There may be leading zeroes in s.

He can perform the following operations on s:

- Delete any digit of the string for a cost of 1.
- Increase any digit of the string by 1 for a cost of x. This operation does not work on the digit `9`.
- Decrease any digit of the string by 1 for a cost of y. This operation does not work on the digit `0`.
- Reverse the string for a cost of 0.

What is the minimum total cost required to turn the string into `90`?

Note that `090`, `0090`, and so on do not count as `90` (there cannot be leading zeroes in the final string).

In this problem, there are multiple independent test cases. For each test case, you are to compute the minimum total cost required to turn the string into `90`, then output the total cost required over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains three space-separated integers n, x, and y.

The second line of each test case contains the string s of length n.

Output Format

Output the sum of the minimum total cost required to turn the string into `90` over all test cases.

Input Constraints

For all subtasks:

- $1 \leq n \leq 16\,000$
- $1 \leq x, y \leq 10\,000$
- s will consist only of numeric digits.

Subtask 1: $tc = 3$, $x = 0$, and $y = 0$

Subtask 2: $tc = 3$, $n = 10$, $x = 1$, and $y = 1$

Subtask 3: $tc = 5$

Sample Testcase

Input	Output
3 2 0 0 09 10 1 1 1234556789 10 1101 1998 2718281828	3116

Sample Testcase Explanation

In the first sample test case, there are $tc = 3$ sub-test cases:

1. In the first sub-test case, you need only reverse the string to get $\texttt{90}$, for a cost of 0.
2. In the second sub-test case, you spend $y = 1$ cost to decrease the first digit $\texttt{1}$ to $\texttt{0}$. Then, we remove the middle 8 digits, and reverse the string. The minimum cost is $1 + 8 = 9$.
3. In the third sub-test case, it can be proven that the minimum cost is 3107.

Hence, output $0 + 9 + 3107 = 3116$.

Input File Link

Link to the Input Files

Task 11: `Fisherman`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133)

Statement

There is a straight river, modelled by the integer positions $1, 2, \ldots, l$ arranged in a line. There are n fish swimming in the river.

Ryan the fisherman reaches this river at time $t = 0$, and sees that the i-th fish is at position $p[i]$ and is swimming to the right.

During each unit of time, every fish moves as follows:

- If a fish is at position $p < l$, then after one time unit it moves to position $p + 1$.
- If a fish is at position $p = l$, then after one time unit it teleports to position 1.

Ryan attempts to catch fish q times. In the j-th attempt, he places a net at time $t[j]$ and at position $x[j]$. This attempt is considered successful if there is at least one fish occupying position $x[j]$ at time $t[j]$. Attempts do not remove fish; each attempt is hypothetical and independent of each other.

Help Ryan find out: how many of the q attempts will be successful?

In this problem, there are multiple independent test cases. For each test case, you are to compute how many of the q attempts are successful, then output the total number of successful attempts over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers n and l.

The second line of each test case contains n space-separated integers $p[1]$, $p[2]$, $\ldots$, $p[n]$, describing the initial positions of the fish.

The third line of each test case contains one integer q.

The following q lines of each test case each contain two space-separated integers. The j-th of these lines contains $t[j]$ and $x[j]$.

Output Format

Output the total number of successful attempts over all test cases.

Input Constraints

For all subtasks, $1 \leq p[i] \leq l$ for all $1 \leq i \leq n$ and $1 \leq x[j] \leq l$ for all $1 \leq j \leq q$.

Subtask 1: $tc = 1$, $n = 3$, $l = 7$, $q = 4$, and $\Sigma\ t[i] = 49$

Subtask 2: $tc = 1$, $n = 100$, $l = 500$, $q = 1000$, and $\Sigma\ t[i] = 100000$

Subtask 3: $tc = 2$, $n = 100000$, $l = 500000$, $q = 200000$, and $\Sigma\ t[i] = 10^{15}$

$\Sigma\ t[i]$ denotes the sum of $t[i]$ across any one test case in each subtask.

Sample Testcase

Input	Output
1 2 5 2 4 3 1 4 4 1 4 2	1

Sample Testcase Explanation

In the first sample test case, $tc = 1$. In the first and only sub-test case, There are $n = 2$ fish in a river of length $l = 5$. The fish are initially at positions $p[1] = 2$ and $p[2] = 4$.

In the first attempt, there will not be a fish at position $x[1] = 4$ at time $t[1] = 1$.

In the second attempt, the fish initially at position 2 will be at position $x[2] = 1$ at time $t[2] = 4$, so Ryan's attempt is successful.

In the third attempt, there will not be a fish at position $x[3] = 2$ at time $t[3] = 4$.

Hence, Ryan has a total of 1 successful attempt.

Input File Link

Link to the Input Files

Task 12: `Pringles Sort`

Authored by: Sun Beichen (TheRaptor)

Prepared by: Sun Beichen (TheRaptor)

Statement

You have a stack of n Pringles chips. Each chip has a unique label from 1 to n. The i-th chip from the bottom has label $a[i]$.

In one move, you can choose any chip from anywhere in the stack and move it to the top of the stack (preserving the relative order of the remaining chips).

What is the minimum number of moves required to sort the stack so that the labels of the chips are in ascending order from bottom to top?

Input Format

The first line of input contains one integer n.

The second line of input contains n space-separated integers $a[1]$, $a[2]$, $\ldots$, $a[n]$, indicating the labels of the chips from bottom to top.

Output Format

Output the minimum number of moves required to sort the stack so that the labels of the chips are in ascending order from bottom to top.

Input Constraints

For all subtasks, $1 \leq a[i] \leq n$ for all $1 \leq i \leq n$ and $a[i] \neq a[j]$ for all $i \neq j$.

Subtask 1: $n = 6$

Subtask 2: $n = 3000$

Subtask 3: $n = 1\,000\,000$

Sample Testcase

Input	Output
5 3 2 1 4 5	4

Sample Testcase Explanation

In the first sample test case, we can first move the chip labelled 2 to the top, then the chip labelled 3, then the chip labelled 4, and finally the chip labelled 5. The final stack will be [1, 2, 3, 4, 5] from bottom to top. It can be shown to be impossible to achieve the objective in less than 4 moves. Hence, the output is 4.

Input File Link

Link to the Input Files

Task 13: `Bamboo`

Authored by: Lim Chang Jun (lcjly)

Prepared by: Brian Lee (penguin133)

Statement

There are n bamboos in a garden. The i-th bamboo has a height of $h[i]$ metres. In one operation, Bob the gardener can magically extend the height of a single bamboo by k metres, where k is a given integer.

Bob the gardener despises bamboos of different heights. He wishes to make the garden have as few distinct bamboo heights as possible.

Among all final height configurations Bob can reach by applying operations, let d_{min} be the smallest possible number of distinct heights. Bob wonders, what is the minimum number of operations required to reach some configuration with d_{min} distinct heights?

In this problem, there are multiple independent test cases. For each test case, you are to compute the minimum number of operations required, then output the total number of operations required over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers n and k.

The second line of each test case contains n space-separated integers $h[1]$, $h[2]$, ..., $h[n]$, describing the initial heights of the bamboos.

Output Format

Output the total number of operations required over all test cases.

Input Constraints

For all subtasks, $1 \leq h[i] \leq 10^9$ for all $1 \leq i \leq n$.

Subtask 1: $tc = 3$, $n = 3$, and $k \leq 40$

Subtask 2: $tc = 2$, $n = 10000$, and $k = 1$

Subtask 3: $tc = 5$, $n = 10000$, and $k \leq 10000$

Sample Testcase

Input	Output
2 2 2 2 6 2 3 1 5	2

Sample Testcase Explanation

In the first sample test case, there are two sub-test cases:

- In the first sub-test case, $n = 2$ and $k = 2$. Bob can apply 2 operations on the first bamboo to extend its height to 6 metres, achieving $d_{min} = 1$.
- In the second sub-test case, $n = 2$ and $k = 3$. Bob can never decrease the number of distinct heights of bamboo, so he will not perform any operations.

Hence, output $2 + 0 = 2$.

Input File Link

Link to the Input Files

Task 14: `E-sign`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133)

Statement

Ryan is a pitiful Admin Support Assistant. He needs n people to e-sign (electronically sign) the same copy of a document. The n people are numbered from person 1 to person n.

This process is modelled as a sequence of $2n$ events, given in chronological order. The i-th event is described by two integers $x[i]$ and $y[i]$, where:

- $x[i]= 0$ means that Ryan sends the document to person $y[i]$ for the first time, and
- $x[i]= 1$ means that person $y[i]$ signs the document and returns it to Ryan.

There are exactly n events with $x = 0$, and exactly n events with $x = 1$. Each person y has exactly one corresponding event with $x = 0$, and exactly one corresponding event with $x = 1$; the event with $x = 0$ appears before the event with $x = 1$.

Whenever a person y signs and returns a copy to Ryan (an event with $x = 1$), Ryan now has an updated copy that includes y's signature. Any person who currently holds an older copy (a person who has already received a copy earlier) but who has not yet signed now has an outdated copy. Immediately after the signing event, Ryan re-sends the updated copy to each of those people who hold an outdated copy. These re-sends happen instantly and are not listed among the input events.

Let $a[j]$ be the total number of times Ryan sends the document to person j. This includes:

- the initial send (the $x = 0$ event for person j), and
- any implicit re-sends that occur after other people sign while person j holds an older copy.

Ryan needs your help to compute the values of $a[1]$, $a[2]$, $\ldots$, $a[n]$ as described. Then, output the weighted sum $a[1] \times 1 + a[2] \times 2 + \cdots + a[n] \times n$.

Input Format

The first line of input contains one integer n.

The following $2n$ lines of input each contain two space-separated integers. The i-th of these lines contains $x[i]$ and $y[i]$.

Output Format

Output the weighted sum $a[1] \times 1 + a[2] \times 2 + \cdots + a[n] \times n$.

Input Constraints

For all subtasks:

- $x[i] \in 0, 1$ for all $1 \leq i \leq 2n$
- $1 \leq y[i] \leq n$ for all $1 \leq i \leq 2n$
- Each person y appears in the input exactly once with $x = 0$ and exactly once with $x = 1$, for all $1 \leq y \leq n$.
- It is guaranteed that the event $0\ y$ appears before the event $1\ y$ for all $1 \leq y \leq n$.

Subtask 1: $n = 6$

Subtask 2: $n = 2000$

Subtask 3: $n = 200\,000$

Sample Testcase

Input	Output
4 0 4 0 1 1 4 0 3 0 2 1 1 1 2 1 3	19

Sample Testcase Explanation

In the first sample test case, Ryan first sends the document to persons 4 and 1.

When person 4 signs the document, Ryan has to re-send the document to person 1 again.

Then, Ryan sends the document to persons 3 and 2.

When person 1 signs the document, Ryan has to re-send the document to persons 3 and 2.

Finally, when person 2 signs, Ryan has to re-send the document to person 3.

This means that $a = [2, 2, 3, 1]$. The output is $2 \times 1 + 2 \times 2 + 3 \times 3 + 1 \times 4 = 19$.

Input File Link

Link to the Input Files

Task 15: `Investment`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Kang Yiming (kym2006) and Brian Lee (penguin133)

Statement

You currently have d dollars, and you intend to invest the dollars over the next n months.

Your monthly expenditure is x dollars. You will set aside x dollars to spend at the start of the month, and invest the rest. The investment rate is r, meaning for every dollar you put in, you get back r dollars at the end of the month.

You want to retire after n months, and you need t dollars to retire. You realise you have fallen short of your target. So, you will choose some months to be thrifty, and reduce your monthly expenditure from x dollars to y dollars. This also means you will invest more money that month.

What is the fewest number of months you need to be thrifty and reduce your expenditure for?

If you don't have enough money to pay that month's expenditure, you simply use all the money you have (your balance will never be negative). This also means that you will invest 0 dollars that month.

If this goal is impossible even if you are thrifty for all n months, output `-1` instead.

Input Format

The first line of input contains six space-separated integers n, d, x, y, r, and t.

Output Format

Output the fewest number of months in which you must be thrifty in order to have at least t dollars after n months; output `-1` if this is impossible.

Input Constraints

Subtask 1: $n = 3$, $d = 100$, $x = 80$, $y = 65$, $r = 3$, and $t = 200$

Subtask 2: $n = 50$, $d = 1$, $x = 2\,000\,000$, $y = 0$, $r = 2$, and $t = 2\,000\,000$

Subtask 3: $n = 30$, $d = 10$, $x = 10^8$, $y = 4$, $r = 2$, and $t = 10^9$

Sample Testcase

Input	Output
3 10 8 6 4 100	2

Sample Testcase Explanation

In the above example, you start with 10 dollars and have 3 months to get 100 dollars. if you decide to be thrifty in the first and third month, your money will change as follows:

1. In the first month, you will spend 6 dollars, leaving yourself with 4 dollars. You invest these 4 dollars, getting back 16 dollars.
2. In the second month, you will spend 8 dollars, leaving yourself with 8 dollars. You invest these 8 dollars, getting back 32 dollars.
3. In the third month, you will spend 6 dollars, leaving yourself with 26 dollars. You invest these 26 dollars and get back 104 dollars, achieving your goal of 100 dollars.

It can be proven that no matter which month is chosen, you will be unable to hit 100 dollars by only being thrifty on one month or less.

Input File Link

Link to the Input Files

Task 16: `Bamboo 2`

Authored by: Ernest Kiew (Shor the Duck)

Prepared by: Ernest Kiew (Shor the Duck)

Statement

Bamboo grows really quickly!

There are n stalks of bamboo, with the i-th stalk having an initial height of $a[i]$ metres. The bamboo will grow for k days, where k is a given integer.

Every day, the following happens:

1. The height of each bamboo stalk doubles; $a[i] := 2a[i]$ for all $1 \leq i \leq n$.
2. If the height of bamboo i is at least $b[i]$ metres, it is cut to a height of $c[i]$ metres.

Let $d[i]$ be the maximum height (in metres) achieved by the i-th stalk of bamboo across the k days (this includes the height right after any doubling, before a same-day cut). What is the sum $d[1] + d[2] + \cdots + d[n]$?

Input Format

The first line of input contains two space-separated integers n and k.

The second line of input contains n space-separated integers $a[1]$, $a[2]$, $\ldots$, $a[n]$, describing the initial height of each stalk of bamboo.

The third line of input contains n space-separated integers $b[1]$, $b[2]$, $\ldots$, $b[n]$.

The fourth line of input contains n space-separated integers $c[1]$, $c[2]$, $\ldots$, $c[n]$.

Output Format

Output the sum $d[1] + d[2] + \cdots + d[n]$.

Input Constraints

For all subtasks:

- $1 \leq k \leq 10^9$
- $1 \leq a[i], b[i], c[i] \leq 10^9$ for all $1 \leq i \leq n$
- $c[i] < b[i]$ for all $1 \leq i \leq n$
- $a[i] < b[i]$ for all $1 \leq i \leq n$

Subtask 1: $n = 1$

Subtask 2: $n = 1000$ and $k = 13$

Subtask 3: $n = 100\,000$

Sample Testcase

Input	Output
5 5 1 1 5 9 3 8 8 7 3200 8 4 5 4 100 5	328

Sample Testcase Explanation

The second bamboo stalk's heights on each day are as follows:

1. On day 1, the bamboo stalk's height doubles from 1 to 2.
2. On day 2, the bamboo stalk's height doubles from 2 to 4.
3. On day 3, the bamboo stalk's height doubles from 4 to 8. Since it is greater than or equal to $b[2] = 8$ metres, its height is cut to 5 metres.
4. On day 4, the bamboo stalk's height doubles from 5 to 10. Since it is greater than or equal to $b[2] = 8$ metres, its height is cut to 5 metres.

5. On day 5, the bamboo stalk's height doubles from 5 to 10. Since it is greater than or equal to $b[2] = 8$ metres, its height is cut to 5 metres.

Thus, the maximum height of the second bamboo stalk is 10 metres.

The maximum heights the bamboo stalks reach are 8, 10, 10, 288, and 12 respectively, summing to 328.

Input File Link

Link to the Input Files

Task 17: `Compressor`

Authored by: Chua Wee Chong (sheep)

Prepared by: Lim Chang Jun (lcjly)

Statement

You are given a binary string (which is a string consisting only of the characters 0 and 1) with **odd** length n. There is a machine called the Compressor that takes a binary string as input and produces a new binary string as output. Let the input string be a, and let the output string be b. The output string b is always exactly two characters shorter than a.

To construct b, the Compressor looks at every substring of length 3 (which is a sequence of three consecutive characters) in a, taken from left to right. For example, the substrings of length 3 in 0110101 are 011, 110, 101, 010, and 101. Notably, substrings can have overlapping characters with each other. For each such substring from left to right, the Compressor determines the most frequent character inside the substring and appends that character to b. Repeating this for all such substrings of a produces the full string b.

You are given an initial binary string s of length n. You will repeatedly apply the Compressor to s: Each time, you replace s with the output produced by the Compressor. The string s will eventually become a single character, as each time the compressor is applied, the length of s shortens by two. Your task is to determine what that final remaining character is.

Input Format

The first line of input will contain tc, the number of testcases.

The first line of each testcase will contain the integer n.

The next line of each testcase will contain the binary string s.

Output Format

Output a string of length tc, where the i-th character represents the answer to the i-th testcase.

Input Constraints

For all subtasks, $1 \leq tc \leq 5$ and n is odd.

Subtask 1: $3 \leq n \leq 5000$

Subtask 2: $3 \leq n \leq 10^6$ and there are only two `'0'`s in s

Subtask 3: $3 \leq n \leq 10^6$

Sample Testcase

Input	Output
1 7 1101010	1

Sample Testcase Explanation

The sequence of transformations is as follows: $1101010 \rightarrow 11010 \rightarrow 110 \rightarrow 1$

Input File Link

Link to the Input Files

Level 3

Task 18: `Jokers`

Authored by: Ernest Kiew (Shor the Duck)

Prepared by: Ernest Kiew (Shor the Duck)

Statement

Shor the Duck is playing Balatro!

In Balatro, a score is equal to the total `chips` multiplied by the total `mult`.

There are n `chips` jokers and m `mult` jokers. Initially, Shor has 1 total `chips` and 1 total `mult`.

The i-th `chips` joker adds $a[i]$ `chips` to the total `chips` when taken, and the j-th `mult` joker adds $b[j]$ `mult` to the total `mult` when taken. Each joker can only be taken at most once.

Shor needs to achieve a minimum of x score to win the current level.

Help Shor determine the fewest jokers he needs to take to still be able to win, or output `-1` if Shor cannot win regardless of which jokers he takes.

In this problem, there are multiple independent test cases. For each test case, you are to compute how many jokers are required or `-1` if it is not possible. Let $ans[k]$ be the answer to the k-th test case. Output $ans[1] \times 1 + ans[2] \times 2 + \cdots + ans[tc] \times tc$.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains three space-separated integers n, m and x.

The second line of each test case contains n space-separated integers $a[1]$, $a[2]$, ..., $a[n]$.

The third line of each test case contains m space-separated integers $b[1]$, $b[2]$, ..., $b[m]$.

Output Format

Output $ans[1] \times 1 + ans[2] \times 2 + \cdots + ans[tc] \times tc$.

Input Constraints

For all subtasks:

- $1 \leq tc \leq 50$
- $1 \leq n, m \leq 1\,000\,000$
- $1 \leq x \leq 10^{18}$
- $1 \leq a[i], b[j] \leq 10^6$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$
- $\Sigma\ a[i] \leq 10^9$ for each test case
- $\Sigma\ b[j] \leq 10^9$ for each test case

Subtask 1: $m = 1$, $b[1] = 1$

Subtask 2: $n, m \leq 1\,000$, $\Sigma n \leq 10\,000$ and $\Sigma m \leq 10\,000$

Subtask 3: $n, m \leq 1\,000\,000$, $\Sigma n \leq 10\,000\,000$ and $\Sigma m \leq 10\,000\,000$

Σn denotes the sum of n over all test cases in a subtask.

Σm denotes the sum of m over all test cases in a subtask.

Sample Testcase

Input	Output
5 5 4 11000 45 31 74 75 35 56 32 53 11 5 4 40000 45 31 74 75 35 56 32 53 11 5 4 38000 45 31 74 75 35 56 32 53 11 5 4 34000 45 31 74 75 35 56 32 53 11 5 4 1 45 31 74 75 35 56 32 53 11	61

Sample Testcase Explanation

In the sample test case, there are $tc = 5$ sub-test cases, all with the same n, m, a, and b:

1. In the first sub-test case, the answer is 4 as you can take the 75 and 74 chip jokers and the 56 and 53 mult jokers, to get $1 + 75 + 74 = 150$ chips and $1 + 56 + 53 = 110$ mult, for a total of $150 \times 110 = 16500$, exceeding 11000 and hence passing the level. It can be proven that 3 jokers is insufficient to pass the level.
2. In the second sub-test case, no matter which jokers Shor takes, he cannot pass the level, and thus the answer is -1.
3. In the third sub-test case, it can be proven that the minimum number of jokers is 9.
4. In the fourth sub-test case, it can be proven that the minimum number of jokers is 8.
5. In the fifth sub-test case, it can be proven that the minimum number of jokers is 0, as Shor can choose to take none of the jokers.

Hence, output $4 \times 1 - 1 \times 2 + 9 \times 3 + 8 \times 4 + 0 \times 5 = 61$.

Input File Link

Link to the Input Files

Task 19: `Musical Chairs`

Authored by: Chua Wee Chong (sheep)

Prepared by: Ryan Goh (rgca) and Chua Wee Chong (sheep)

Statement

There are n chairs in a row, labelled 1 to n from left to right.

There are also m people, labelled from 1 to m. The i-th person initially sits on chair $c[i]$.

Each person has a direction $d[i]$ that is either `L` or `R`:

- If person p has direction `L`, they are allowed to move one chair to the left (chair x to chair $x - 1$) if that chair is empty.
- Similarly, if person p has direction `R`, they are allowed to move one chair to the right (chair x to chair $x + 1$) if that chair is empty.

You are in charge of moving people. What is the maximum number of moves you can perform?

In this task, there are multiple independent test cases. For each test case, you are to compute the maximum number of moves you can perform, then output the sum of the maximum number of moves over all test cases.

Input Format

The first line of input contains one integer t, the number of test cases.

The first line of each test case contains two space-separated integers n and m.

The following m lines of each test case each contain a character and an integer, separated by a space. The i-th of these lines contains $d[i]$ and $c[i]$.

Output Format

Output the sum of the maximum number of moves over all test cases.

Input Constraints

For all subtasks:

- $1 \leq tc \leq 5$
- $1 \leq m \leq n \leq 10^6$
- $d[i] \in$ `L`, `R` for all $1 \leq i \leq m$
- $1 \leq c[i] \leq n$, $c[i] \neq c[j]$ if $i \neq j$ for all $1 \leq i \leq m$

Subtask 1: $d[i] =$ `L` for all $1 \leq i \leq n$

Subtask 2: All persons with direction `R` are initially to the left of all persons with direction `L`

Subtask 3: $1 \leq n \leq 10^6$

Sample Testcase

Input	Output
3 51 6 L 9 L 11 L 20 L 23 L 34 L 51 12 2 R 6 L 7 12 5 L 2 R 3 R 6 L 8 R 10	134

Sample Testcase Explanation

In the first sample test case, there are $tc = 3$ sub-test cases:

1. In the first sub-test case, the maximum number of moves is 127.
2. In the second sub-test case, the person on chair 6 wants to move right but the person on chair 7 wants to move left. Hence there is no way for any of them to move, so maximum number of moves is 0.
3. In the third sub-test case, it can be proven that the maximum number of moves is 7.

Hence, output $127 + 0 + 7 = 134$.

Input File Link

Link to the Input Files

Task 20: `Volcano`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Zhao Yaoqi (zyq1810)

Statement

There is an ocean which can be modelled as a $l \times l$ grid of cells. The rows of the grid are numbered 1 to l from north to south, and the columns of the grid are numbered 1 to l from west to east. We refer to the cell located at row x and column y of the grid as cell (x, y).

Initially, every cell is water, except for n volcano cells. The i-th volcano is located at ($r[i]$, $c[i]$) and contains magma of height $h[i]$.

Suddenly, all the volcanoes erupt at once. The magma spreads in discrete seconds; the following happens each second:

- If a cell currently has magma of height $h \geq 1$, then every water cell or magma cell with height less than $h - 1$ that is adjacent to it (in any of the 8 directions) becomes a magma cell of height $h - 1$ in the next second.

If several magma sources would create magma in the same water cell in the same step, then (in the next second) the magma height of that cell would be equal to the maximum height it would have received from any individual source.

After a long time, all magma cools into rocks. Cells that never received magma remain water.

An island is a maximal connected group of rock cells, where cells are connected by 8-directional adjacency.

How many islands of rocks will there be in the end?

In this problem, there are multiple independent test cases. For each test case, you are to compute how many islands of rocks there will be in the end, then output the total number of islands of rocks over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers l and n.

The following n lines of each test case each contain three space-separated integers. The i-th of these lines contains $r[i]$, $c[i]$, and $h[i]$.

Output Format

Output the total number of islands of rocks over all test cases.

Input Constraints

For all subtasks, $1 \leq r[i], c[i] \leq l$ for all $1 \leq i \leq n$ and $0 \leq h[i] \leq l - 1$ for all $1 \leq i \leq n$.

Subtask 1: $tc = 3$, $1 \leq l \leq 20$, and $1 \leq n \leq 6$

Subtask 2: $tc = 10$, $1 \leq l \leq 1000$, and $1 \leq n \leq 30$

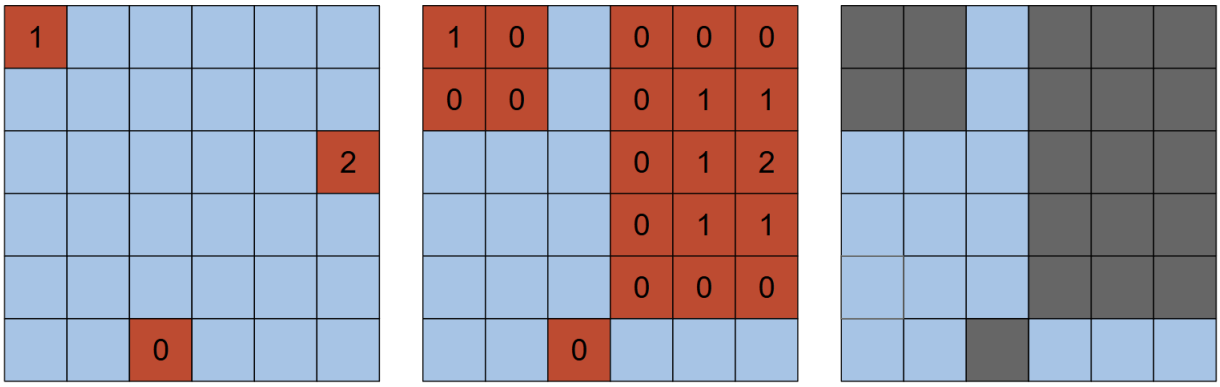
Subtask 3: $tc = 10$, $1 \leq l \leq 10^9$, and $1 \leq n \leq 5000$

Sample Testcase

Input	Output
1 6 3 6 3 0 1 1 1 3 6 2	2

Sample Testcase Explanation

Visualisation for sample testcase:



There are two islands as the rock on row 5 column 2 is connected to the rock on row 4 column 3.

Input File Link

Link to the Input Files

Task 21: `Card Draw`

Authored by: Ernest Kiew (Shor the Duck) & Brian Lee (penguin133)

Prepared by: Zhao Yaoqi (zyq1810)

Statement

There are two parts to this task, specified by an integer $\mathbf{p}$ in the input for each test case. This will only affect the second step in each turn.

Shor the Duck is playing a card game! There are two piles of cards: one is the draw pile, and the other is in Shor's hand. Each card has an integer value labelled on it.

Initially, there are n cards in the draw pile, with the i-th card from the top having the value $a[i]$ labelled on it. Shor's hand is empty at the start.

Shor would start the game by taking the topmost card and putting it in his hand. Then, he would repeatedly take turns to play. In each turn, the following steps happen in order:

1. Shor chooses a card from his hand to play. Suppose the card has the value x.
2. If $\mathbf{p} = \mathbf{1}$, Shor will put the card back in his hand. If $\mathbf{p} = \mathbf{2}$, Shor will put the card at the bottom of the draw pile.
3. Shor draws x cards from the top of the deck into his hand, or until the deck is empty if there are less then x cards in the deck.
4. If the pile is empty, Shor wins.

It is guaranteed that Shor can win in a finite number of steps for given inputs.

What is the minimum number of turns that Shor needs to take to win?

In this task, there are multiple independent test cases. For each test case, you are to compute what is the minimum number of turns needed for Shor to win, then output the total number of turns over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers p and n.

The following line of each test case each contains n space-separated integers. The i-th of these integers represents $a[i]$.

Output Format

Output the total number of turns over all test cases.

Input Constraints

For all subtasks, $1 \leq a[i] \leq n$ for all $1 \leq i \leq n$

Subtask 1: $tc = 10$, $p = 1$, and $2 \leq n \leq 2 \times 10^5$

Subtask 2: $tc = 10$, $p = 2$, and $2 \leq n \leq 1000$

Subtask 3: $tc = 10$, $p = 2$, and $2 \leq n \leq 2 \times 10^5$

Sample Testcase

Input	Output
2 1 6 2 1 1 3 2 1 2 6 2 1 1 3 2 1	8

Sample Testcase Explanation

For the first test case, Shor can do the following:

1. Shor's hand: 2. Deck: 1, 1, 3, 2, 1. Shor plays 2.

2. Shor's hand: 2, 1, 1. Deck: 3, 2, 1. Shor plays 2.
3. Shor's hand: 2, 1, 1, 3, 2. Deck: 1. Shor plays 2.
4. Shor's hand: 2, 1, 1, 3, 2, 1. Shor wins!

For the second test case, Shor can do the following:

1. Shor's hand: 2. Deck: 1, 1, 3, 2, 1. Shor plays 2.
2. Shor's hand: 1, 1. Deck: 3, 2, 1, 2. Shor plays 1.
3. Shor's hand: 1, 3. Deck: 2, 1, 2, 1. Shor plays 3.
4. Shor's hand: 1, 2, 1, 2. Deck: 1, 3. Shor plays 2.
5. Shor's hand: 1, 1, 2, 1, 3. Deck: 2. Shor plays 3.
6. Shor's hand: 1, 1, 2, 1, 2, 3. Shor wins!

This gives a total of 8 turns. It can be proved that this is the minimum possible total number of turns.

Input File Link

Link to the Input Files

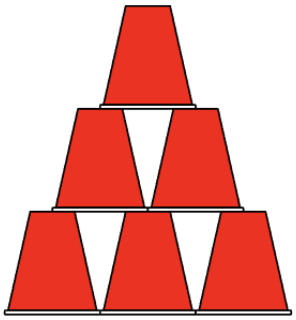
Task 22: `Cup Stacking`

Authored by: Sun Beichen (TheRaptor)

Prepared by: Zhao Yaoqi (zyq1810)

Statement

There is a pyramid of cups. The pyramid has n layers and the i-th layer from the top has i cups. Every cup above the n-th layer is supported by two cups below it as shown in the diagram.



The cups are designed in a way that you can slot one above another to form a cup-stack.

You want to collapse the pyramid to height 1. You can do so by picking a cup/cup-stack x that is not supporting any cup/cup-stack, then slotting it above one of the cups that is supporting x but is not supporting any other cup/cup-stack. This effectively combines them into a single cup-stack.

The process ends when no cup is supported by any other cup. At the end, you will end up with n stacks of cups on the table. For the stack in i-th position from the left, each cup in the stack increases your score by $a[i]$.

Given the values $a[i]$, what is the maximum possible score you can achieve?

In this task, there are multiple independent test cases. For each test case, you are to compute what is the maximum score you can achieve, then output the total sum of maximum scores over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains one integer n.

The following line of each test case each contains n space-separated integers. The i-th of these integers represents $a[i]$.

Output Format

Output the total sum of maximum scores of over all test cases.

Input Constraints

Subtask 1: $tc = 3$, $1 \leq a[i] \leq 5$ for all $1 \leq i \leq n$, and $1 \leq n \leq 5$

Subtask 2: $tc = 10$, the array a contains one '1' and $n - 1$ '0's, and $1 \leq n \leq 10^6$

Subtask 3: $tc = 10$, $1 \leq a[i] \leq 10^6$ for all $1 \leq i \leq n$, and $1 \leq n \leq 10^6$

Sample Testcase

Input	Output
1 3 1 3 2	14

Sample Testcase Explanation

The following can be done:

1. Stack the cup in the top layer above the left cup in the middle layer.
2. Stack the right cup in the middle layer on the rightmost cup in the bottom layer.

3. Stack the remaining cup-stack of 2 cups in the middle layer on the middle cup in the bottom layer.

This gives a maximum score of 14.

Input File Link

Link to the Input Files

Task 23: `Mildly Angry Ducks`

Authored by: Sun Beichen (TheRaptor)

Prepared by: Chua Wee Chong (sheep)

Statement

There is a row of n ducks, the i-th belonging to nation $d[i]$. When two ducks from different nations are next to each other, they fight, resulting in mutually assured destruction.

On day 0, all n ducks are in their aforementioned positions. For every subsequent day, a duck disintegrates if it was previously next to a duck from a different nation. Refer to the sample for more details.

Eventually, either all ducks disintegrates or only ducks from one nation remains. Compute the number of ducks that survive.

In this problem, there are multiple independent test cases. For each test case, you are to compute the number of ducks that survive, then output the total of them over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains one integer n.

The second line of each test case contains n space-separated integers $d[1]$, $d[2]$, $\ldots$, $d[n]$.

Output Format

Output the total number of ducks that survive after all fighting has ended over all test cases.

Input Constraints

For all subtasks: $1 \leq tc \leq 6$ and $1 \leq d[i] \leq n$ for all $1 \leq i \leq n$.

Subtask 1: $1 \leq n \leq 5000$

Subtask 2: $1 \leq n \leq 2 \times 10^6$ and if $d[i] = d[j]$ and $i < j$, then all $d[i]$, $d[i+1]$, $\ldots$, $d[j]$ are equal to $d[i]$

Subtask 3: $1 \leq n \leq 2 \times 10^6$

Sample Testcase

Input	Output
2 9 1 2 2 2 2 3 1 3 3 6 1 1 1 2 2 2	1

Sample Testcase Explanation

In the first test case of the sample, on day 0, the position of ducks are $[1, 2, 2, 2, 2, 3, 1, 3, 3]$.

On day 1, the position of ducks are $[2, 2, 3]$.

On day 2, the position of ducks are $[2]$.

Hence, the answer to the first test case is 1.

In the second test case of the sample, the answer is 0.

Input File Link

Link to the Input Files

Task 24: `Tick Tock`

Authored by: Lim Chang Jun (lcjly)

Prepared by: Ryan Goh (rgca)

Statement

Bob has a special clock. Unlike a normal clock with 12 numbers, this clock has n numbers arranged in a circle, each one equally spaced apart just like a normal clock.

m monsters will spawn one by one at positions on the clock (so there will only be at most one monster at any given time). The i-th monster will spawn on top of number $a[i]$. Consecutive monsters never spawn on the same number.

The clock has two hands: the hour hand, the minute hand, both simultaneously handled by the clock's crown.

When Bob rotates the crown one full rotation in a specified direction (clockwise or anticlockwise), the minute hand will move to the next number in the specified direction, whereas the hour hand will only move $\frac{1}{n}$ of the distance covered by the minute hand in the specified direction.

A monster is eliminated if either the minute hand or the hour hand touches its position **exactly**. Once the i-th monster is eliminated, the $i+1$-th monster, if any will immediately spawn in its specified position.

Initially, both hands point at number n. Bob wants to eliminate all monsters in the given order. Compute the minimum number of full rotations of the crown required to eliminate all monsters.

In this problem, there are multiple independent test cases. For each test case, you are to compute the minimal number of full rotations required. Let $ans[k]$ be the answer to the k-th test case. Output $ans[1]\times 1 + ans[2]\times 2 + \cdots + ans[tc]\times tc$.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers n and m.

The second line of each test case contains m space-separated integers $a[1], a[2], \ldots, a[m]$.

Output Format

Output $ans[1]\times 1 + ans[2]\times 2 + \cdots + ans[tc]\times tc$.

Input Constraints

For all subtasks, $1 \leq tc \leq 10$ and $1 \leq a[i] \leq n$ for all $1 \leq i \leq m$

Subtask 1: $n = 3$, $m \leq 10$

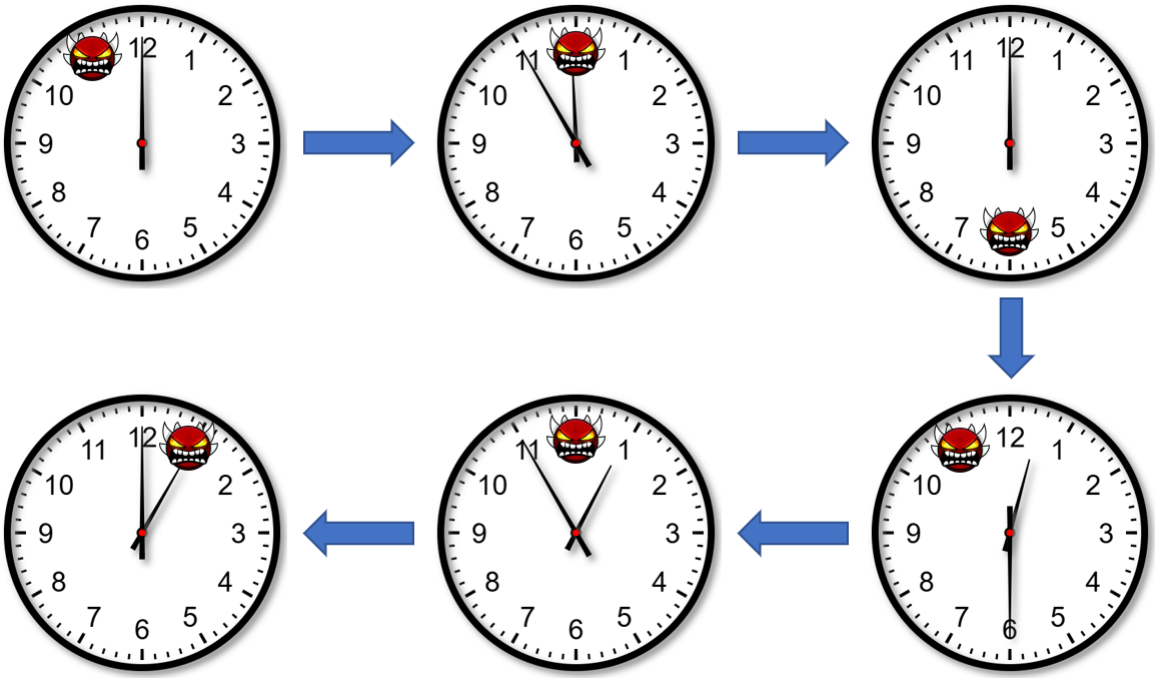
Subtask 2: n, $m \leq 500$

Subtask 3: $n \leq 5000$, $m \leq 500$

Sample Testcase

Input	Output
3 12 6 11 12 6 11 12 1 3 6 2 1 2 3 2 1 516 9 299 435 67 1 332 2 516 383 1	3643

Sample Testcase Explanation



The first sample testcase describes a case of a normal analog clock (with $n = 12$). It starts at 12.00 am, and a full rotation of the crown in this problem only moves the minute hand by 5 minutes.

The first monster spawns at number 11, and so we turn the crown one full rotation clockwise such that the minute hand moves from 12 to 11. We kill the monster at number 11, and immediately the next monster spawns at number 12.

We then turn the crown clockwise to adjust the minute hand to 12, 6, 11 and 12 respectively, killing the 2nd to 5th monsters. This takes 13 clockwise rotations of the crown.

Finally, at 1.00 am, since the hour hand is exactly at number 1, we can also kill the monster at number 1 too with no additional cost. In total we need 14 full rotations of the crown to kill all the monsters, and this is the minimum possible.

The second testcase has answer 4 and the 3rd testcase has answer 1207. Hence the output is $14 \times 1 + 4 \times 2 + 1207 \times 3 = 3643$.

Input File Link

Link to the Input Files

Task 25: `Mahjong`

Authored by: Brian Lee (penguin133)

Prepared by: Brian Lee (penguin133)

Statement

Brian the chronic gambler is now at a mahjong table, ready to win some money! In this game of mahjong, there are n different types of tiles. Brian's hand currently consists of $a[i]$ tiles of type i for each $1 \leq i \leq n$.

At the start of Brian's turn, he will draw a single tile, which can have any type between 1 to n, and add it to his hand. He wins if he is able to group all of his tiles into groups of k, where each group's tile types form a consecutive range of length k.

Help Brian calculate the number of tile types which, if drawn on this turn, allows him to win.

In this problem, there are multiple independent test cases. For each test case, you are to compute the number of different tiles that lets Brian win, then output the total number of such tiles over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers, n and k.

The second line of each test case contains n space-separated integers $a[1]$, $a[2]$, $\ldots$, $a[n]$.

Output Format

Output the total number of tiles that allow Brian to win across all testcases.

Input Constraints

For all subtasks, $tc = 15$, $1 \leq k \leq n \leq 4 \times 10^5$ and $0 \leq a[i] \leq 10^9$ for all $1 \leq i \leq n$.

Subtask 1: $n \leq 6000$ and $\Sigma n = 30000$.

Subtask 2: $\Sigma n \leq 2 \times 10^6$ and $k \leq 10$.

Subtask 3: $\Sigma n \leq 2 \times 10^6$.

Σn denotes the sum of n over all test cases in a subtask.

Sample Testcase

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

Sample Testcase Explanation

For the first testcase, Brian can draw either tile 2, tile 4 or tile 6 to win.

For the second testcase, Brian can only draw tile 3 to win.

For the third testcase, it can be shown that Brian will not win no matter which tile he draws.

Input File Link

Link to the Input Files

Task 26: `Slimes`

Authored by: Sun Beichen (TheRaptor)

Prepared by: Chua Wee Chong (sheep)

Statement

There are n slimes in a row. Slime i has size $s[i]$. Slimes can eat other slimes either to their left or to their right, that is, adjacent slimes.

When slime a of size x eats slime b of size y, slime a becomes of size $x + y$ and slime b disappears. Then, the other slime to the adjacent to slime b would now be adjacent to slime a instead.

Let $a[i]$ be the minimal number of times slime i must eat other slimes so that no slime has a larger size than it. Find the sum $a[1] + a[2] + \cdots + a[n]$.

In this problem, there are multiple independent test cases. For each test case, you are to compute the sum $a[1] + a[2] + \cdots + a[n]$, then output the total of the sums over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains one integer n.

The second line of each test case contains n space-separated integers $s[1]$, $s[2]$, ..., $s[n]$.

Output Format

Output the total of the sums over all test cases.

Input Constraints

For all subtasks, $1 \leq tc \leq 5$ and $1 \leq s[i] \leq 10^9$ for all $1 \leq i \leq n$.

Subtask 1: $1 \leq n \leq 500$

Subtask 2: $1 \leq n \leq 10000$

Subtask 3: $1 \leq n \leq 10^6$

Sample Testcase

Input	Output
2 5 1 2 3 4 1 5 1 1 1 1 2	9

Sample Testcase Explanation

In the first test case of the sample:

For the first slime, it will first eat slime 2 and become size $1 + 2 = 3$. Then, it is adjacent to slime 3 and eats slime 3 and becomes size $3 + 3 = 6$. There is no slime larger than slime 1 after that. Slime 1 will eat 2 slimes.

For the second slime, it will eat slime 3 and become size $2 + 3 = 5$. There is no slime larger than slime 2 after that. Slime 2 will eat 1 slime.

For the third slime, it will eat slime 4 and become size $3 + 4 = 7$. There is no slime larger than slime 3 after that. Slime 3 will eat 1 slime.

For the fourth slime, there is no slime larger than slime 4. Slime 4 will eat 0 slimes.

For the fifth slime, it will eat slime 4 and become size $4 + 1 = 5$. There is no slime larger than slime 5 after that. Slime 5 will eat 1 slime.

In total, summing the values up, we get the final answer of $2 + 1 + 1 + 0 + 1 = 5$.

In the second test case of the sample, $a = [1, 1, 1, 1, 0]$.

Input File Link

Link to the Input Files

Level 4

Task 27: `Bad Addition`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Chua Wee Chong (sheep)

Statement

Brian has just learnt how to add two numbers! Unfortunately, he has learnt the wrong things.

Normally, when adding two numbers, there might be a carry digit to be added to the next significant place to the immediate left. Instead, Brian will just insert the carry digit into the number.

For example, $98 + 76 = 1614$ for Brian, whereas it would be 174 normally.

Now, Brian has to add a number n to 0. He will do this k times, each time taking the result of the previous addition. Refer to the sample for more details. What number will he end up with? As this number may be very big, output it mod $10^9 + 7$.

Input Format

The first line of input contains two space-separated integers n and k.

Output Format

Output a single integer, the final result after all the additions, modulo $10^9 + 7$

Input Constraints

For all subtasks, $1 \leq n \leq 99999$.

Subtask 1: $k = 10$

Subtask 2: $k = 272727$

Subtask 3: $k = 407600000000000000$

Sample Testcases

Input	Output
78 3	14814

Input	Output
11111 10	10101003

Sample Testcase Explanation

In the first sample, we want to add 78 to 0, doing this 3 times. The final result is 14814. The steps are:

1. $0 + 78 = 78$
2. $78 + 78 = 1416$
3. $1416 + 78 = 14814$

In the second sample, our result is 1010101010. Taking modulo $10^9 + 7$, we get 10101003.

Input File Link

Link to the Input Files

Task 28: `Company Management`

Authored by: Sun Beichen (TheRaptor)

Prepared by: Sun Beichen (TheRaptor)

Statement

There are n employees in a company, indexed from 1 to n. Every employee i, except for employee 1, has a direct superior with index $p[i]$, which is guaranteed to be smaller than i. Employee j is said to be a subordinate of employee i if i can be found in j's chain of direct or indirect superiors (i.e. direct superior, superior's superior, etc).

For each hour of work that an employee i does, the company makes $k[i]$ money, but employee i and all of their subordinates will have their stress level increased by 1.

Regulation states that no employee can have a stress level greater than m. As the CEO, you need to determine the maximum total money the company can make, by assigning non-negative integer work hours to each employee.

In this problem, there are multiple independent test cases. For each test case, you are to compute the maximum total money the company can make, then output the sum of the maximum total money the company can make over all test cases, modulo $10^9 + 7$.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains two space-separated integers n and m.

The second line of each test case contains $n - 1$ space-separated integers $p[2]$, $p[3]$, $\ldots$, $p[n]$.

The third line of each test case contains n space-separated integers $k[1]$, $k[2]$, $\ldots$, $k[n]$.

Output Format

Output the sum of the maximum total money the company can make over all test cases, modulo $10^9 + 7$.

Input Constraints

For all subtasks: $1 \leq tc \leq 10$, $2 \leq n \leq 500000$, $1 \leq m \leq 1000000$, $1 \leq p[i] \leq i$ for all $2 \leq i \leq n$ and $0 \leq k[i] \leq 1000000$ for all $1 \leq i \leq n$.

Subtask 1: $p[i] = 1$ for all $1 \leq i \leq n$

Subtask 2: $m = 1$

Subtask 3: No additional constraints

Sample Testcase

Input	Output
1 5 2 1 1 2 2 4 3 2 1 2	10

Sample Testcase Explanation

For this input, one of the possible distributions is to assign 2 hours of work to employee 3, then 1 hour of work each to employee 2, employee 4, and employee 5. The stress levels faced by the employees (in order of index) are [0, 1, 2, 2, 2] and the total amount of money made is 10. It is impossible to make more money than this while keeping everyone's stress level below or equal to 2.

Input File Link

Link to the Input Files

Task 29: `Mixing Sauces`

Authored by: Lim Rui Yuan (oolimry)

Prepared by: Ryan Goh (rgca)

Statement

A sauce has saltiness a if it has a units of salt per litre of water, and sweetness b if it b units of sugar per litre of water.

You have n types of base sauces, sauce i having saltiness $a[i]$ and sweetness $b[i]$. You have only exactly one of each type of base sauce.

You want to make m new sauces of saltiness $c[j]$ and sweetness $d[j]$, by mixing any amount of the existing sauces in any ratio. You **CANNOT** dilute the sauces with plain water.

For each of the m sauces, determine if it is possible to make it.

Input Format

The first line of the input will contain n and m, the number of base sauces that you have and the number of new sauces that you will like to make.

For the next n lines of input, each line will contain 2 integers: $a[i]$ and $b[i]$ respectively, representing the saltiness and sweetness respectively for base sauce i.

For the next m lines of input, each line will contain 2 integers: $c[j]$ and $d[j]$ respectively, representing the saltiness and sweetness respectively for the j-th sauce that you will like to make.

It is guaranteed that no two base sauces have the exact same saltiness and sweetness.

Output Format

Output a binary string (string containing of only '0's and '1's) of length m. A '0' in the j-th position represents that new sauce j cannot be made, and '1' in the j-th position represents that

new sauce j can be made.

Input Constraints

For all subtasks: $1 \leq a[i], b[i], c[j], d[j] \leq 100$ and if $a[x] = a[y]$ and $b[x] = b[y]$, then $x = y$.

Subtask 1: $n = 2, m = 10$

Subtask 2: $n = 3, m = 6$

Subtask 3: $n = 100, m = 40$

Sample Testcase

Input	Output
5 4 60 100 30 90 25 75 12 21 25 50 18 44 43 84 60 40 53 95	1101

Sample Testcase Explanation

The first new sauce can be made by mixing $\frac{1}{3}$ of the second base sauce with $\frac{2}{3}$ of the fourth base sauce.

The second new sauce can be made by mixing $\frac{1}{2}$ of the first base sauce, $\frac{1}{10}$ of the second base sauce, and $\frac{1}{5}$ of the third and fifth base sauces together.

It can be shown that the third new sauce cannot be made, but the fourth one can.

Input File Link

Link to the Input Files

Task 30: `Mountain Ranges`

Authored by: Yaw Chur Zhe (pavement)

Prepared by: Yaw Chur Zhe (pavement)

Statement

There is a mountain range modelled as a row of n mountains numbered from 1 to n from left to right. Mountain i has height $h[i]$ units. It is guaranteed that no two mountains have the same height.

We say that mountain x can see the top of mountain y if $h[x] > h[y]$ and there does not exist a mountain z between mountains x and y such that $h[z] > h[x]$. Mountain x can see the top of itself.

We call a mountain a vantage point if it can see the tops of strictly more than half of all mountains.

We call a mountain range beautiful if strictly more than half of all mountains are vantage points.

Compute the length of the longest beautiful subsequence of mountains of the original mountain range.

In this problem, there are multiple independent test cases. For each test case, you are to compute the length of the longest beautiful subsequence of mountains of the original mountain range, then output the sum of lengths over all test cases.

Input Format

The first line of input contains one integer tc, the number of test cases.

The first line of each test case contains one integer n.

The second line of each test case contains n space-separated integers $h[1]$, $h[2]$, $\ldots$, $h[n]$.

Output Format

Output the sum of lengths of the longest beautiful subsequence over all test cases.

Input Constraints

For all subtasks:

- $1 \leq h[i] \leq n$ for all $1 \leq i \leq n$
- $h[i] \neq h[j]$ for all $i \neq j$

Subtask 1: $tc = 10$ and $1 \leq n \leq 20$

Subtask 2: $tc = 20$ and $1 \leq n \leq 2000$

Subtask 3: $tc = 20$ and $1 \leq n \leq 200\,000$

Sample Testcase

Input	Output
3 5 3 2 1 5 4 3 2 1 3 7 4 6 2 3 1 5 7	11

Sample Testcase Explanation

In the sample input, there are three test cases:

1. In the first test case, one of the longest beautiful subsequences is 2, 1, 5. Hence, the answer is 3.

2. In the second test case, the original mountain range is already beautiful. Hence, the answer is $n = 3$.
3. In the third test case, one of the longest beautiful subsequences is 4, 2, 3, 1, 5. Hence, the answer is 5.

The output is $3 + 3 + 5 = 11$.

Input File Link

Link to the Input Files

Optimisation Task

Wandering Robot

Authored by: Sun Beichen (TheRaptor)

Prepared by: Sun Beichen (TheRaptor)

Statement

A robot is moving on a grid with n rows and m columns. The rows of the grid are numbered 1 to n from top to bottom, and the columns of the grid are numbered 1 to m from left to right. We refer to the cell located at row r and column c of the grid as cell (r, c).

Each cell of the grid is either:

- empty, or
- contains a wall. Every wall is labelled either `L` or `R`.

The robot begins at (x, y), initially facing right. The robot always moves straight forward in the direction it is currently facing, until one of two things happens:

1. It is about to leave a grid. In this case, the robot stops.
2. It is about to reach a wall. If the wall is labelled `L`, the robot turns left. If it is labelled `R`, the robot turns right. After turning, the robot continues moving forward again.

You are given the starting position of the robot (x, y). Your task is to decide where to place walls (labelled `L` or `R`) in the grid in order to maximise the number of unique empty cells the robot visits. If the robot passes through the same cell multiple times, it only counts once.

You should aim to produce the best possible construction; better outputs receive higher scores. You will still receive partial scores for sub-optimal constructions.

Input Format

The first line of input contains two space-separated integers n and m.

The second line of input contains two space-separated integers x and y.

Output Format

Output a grid with n rows and m columns. Empty cells should be represented with a '`.`' and walls are represented with either '`L`' or '`R`'. The cell (x, y) must be empty.

Your raw score is the number of unique empty cells the robot visits when following the rules described in the problem statement. Cells visited more than once are only counted once.

Sample Testcase

Input	Output
3 3 2 1	... ..R .R.

Sample Testcase Explanation

For this specific output, the robot will start at $(2, 1)$ facing right, move to $(2, 2)$, turn right due to the wall at $(2, 3)$, turn right again due to the wall at $(3, 2)$, move to $(2, 1)$, then finally stop just before it exits the grid. The robot visits 2 unique empty cells.

Although it is possible for the robot to visit more than 2 cells in a 3×3 grid, this output would still receive partial score for visiting 2 cells.

Input File Link

Link to the Input Files