2026 NTU ICPC Team Preliminary

National Taiwan University

2026/08/09

Language	Version	Compile Flags	Extensions
C	gcc 13.3.0	`-x c -Wall -O2 -static -std=gnu17 -lm`	`.c`
C++	g++ 13.3.0	`-x c++ -Wall -O2 -static -std=gnu++23`	`.cc, .cpp`
Python	python 3.9.18	`None`	`.py`

Problem	Problem Name	Time Limit	Memory Limit
A	All-Save Challenge	2 s	2048 MB
B	Buy One Get One Free	2 s	2048 MB
C	Close to Gray	2 s	2048 MB
D	Distinct Numbers	1 s	2048 MB
E	Electric Vehicle Revisited	4 s	2048 MB
F	Flos	1 s	2048 MB
G	Great Cactus Comeback	5 s	2048 MB
H	Hexagon Game	1 s	2048 MB
I	Inversion Counting	5 s	2048 MB
J	Jester's Paint Parade	2 s	2048 MB
K	Kuusi Seitsemän	1 s	2048 MB
L	Largest Prime Factor	2 s	2048 MB
M	Merging Slimes	5 s	2048 MB

This page is intentionally left blank.

A. All-Save Challenge[1] [2]

Problem ID: triangle

After a terrible storm, you receive the locations of n people waiting to be rescued. To protect them until help arrives, you can create a safety zone in the shape of an **equilateral triangle**. The safety zone may be placed anywhere and rotated by any angle. A person is protected if their location is inside the triangle or on its boundary.

Since a larger safety zone requires more energy, you want to make it as small as possible. Find the minimum possible area of a safety zone that protects all n people.

Recall that a triangle is called an equilateral triangle if and only if all its sides have the same length.

Input

The first line contains an integer n.

The i-th line of the next n lines contains two integers x_i and y_i, representing the location of the i-th person.

- $1 \leq n \leq 5 \times 10^5$
- $-10^9 \leq x_i, y_i \leq 10^9$
- No two people are at the same location.

Output

Output the minimum possible area.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

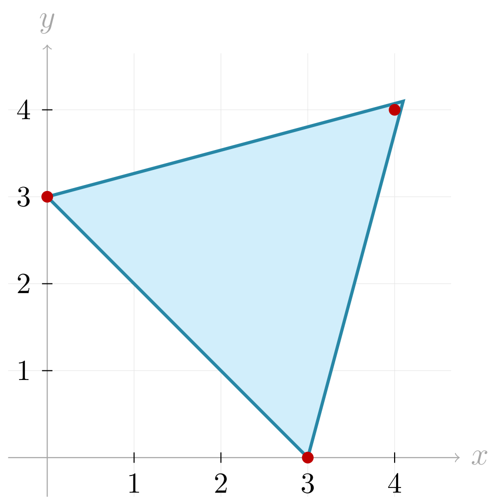
Sample Input 1	Sample Output 1
3 3 0 4 4 0 3	7.79422863405994782087

Sample Input 2	Sample Output 2
5 -1 -2 3 1 0 4 4 3 -3 2	34.02164503538821491556

Sample Input 3	Sample Output 3
2 -1000000000 -1000000000 1000000000 1000000000	3464101615137754587.05489268301174473389

Note

The following figure shows a minimum-area safety zone for the first sample. The red points represent the locations of the three people.



B. Buy One Get One Free

Problem ID: starbucks

Watermelon is a kindergarten teacher who needs to invent games for her energetic kids.

One day, she invented a new game. First, she gathered n kids and drew a circle on the ground. Every kid is out of the circle initially. The i-th kid is assigned a number a_i. She then announced the rules:

- The kids must determine a permutation and submit it to her, she will call the kids one by one according to the permutation.
- Once a kid is called, he will step into the circle. Then, let a_i be the number assigned to this kid, all kids currently in the circle must divide themselves into groups of size a_i. Kids not in any group will be removed out of the circle. The kids can determine who to leave the circle, as long as the remaining kids in the circle can be divided into groups of size a_i.
- The game ends when every kid has been called. A kid who has been removed out of the circle cannot be called again. When the game ends, each kid still in the circle receives a cookie as a reward.

Since the kids are cooperative, they will share the cookie after the game ends. Thus, they only care about the total number of the cookie that watermelon will give out. Please help the kids determine the order in which they should be called so that they obtain the maximum number of cookies.

Note that if a circle contains no kids, it is considered to be divided into groups of size x for any positive integer x.

Input

The first line contains a positive integer n, the number of kids. The second line contains n positive integers $a_1, a_2, \ldots, a_n$, representing the numbers assigned to each kid.

- $1 \leq n \leq 10000$
- $1 \leq a_i \leq n$

Output

Output two lines. The first line is the maximum number of cookies they can receive. The second line is a permutation of size n, representing the order they should be called. The kids are numbered from 1 to n.

Sample Input 1

```
4
1 1 3 4
```

Sample Output 1

```
4
2 1 3 4
```

Sample Input 2

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

Sample Output 2

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

Note

In the second sample, the kids can guarantee to earn 6 cookies if the teacher calls the kids in the following order $1, 2, 3, 4, 7, 8, 5, 6$:

1. In the first four steps, four kids are called and the kids always form exactly one group.
2. In the fifth step, the teacher calls the 7-th kid, since there are 5 kids in the circle currently, 2 kids are removed and the rest of them forms one group of size 3.
3. In the next step, the teacher calls the 8-th kid, all the remaining kids form one group of size 4.
4. In the next step, the teacher calls the 5-th kid, each of the kids forms one group of size 1 and nobody will be removed.
5. In the last step, the teacher calls the 6-th kid, the kids will form three groups of size 2. Since all kids have been called and there are 6 kids remaining in the circle, the teacher gives 6 cookies out.

C. Close to Gray[1] [2]

Problem ID: complement

You are given a simple undirected graph G with n vertices and m black edges. The i-th black edge connects vertices u_i and v_i. For every pair of distinct vertices not connected by a black edge, there is a white edge between them.

We call two distinct vertices x and y *close to gray* if both of the following conditions hold:

- It is possible to travel from x to y using only black edges.
- It is possible to travel from x to y using only white edges.

Find the number of unordered pairs (x, y), where $1 \leq x < y \leq n$, that are *close to gray*.

Input

The first line contains two integers n and m, representing the number of vertices and black edges, respectively.

Each of the next m lines contains two integers u_i and v_i, indicating that there is a black edge between them.

- $2 \leq n \leq 5 \times 10^5$
- $0 \leq m \leq 5 \times 10^5$
- $1 \leq u_i, v_i \leq n$
- $u_i \neq v_i$
- There are no multiple edges between the same pair of vertices.

Output

Output a single integer representing the number of unordered pairs that are *close to gray*.

Sample Input 1	**Sample Output 1**
4 4 1 3 2 3 1 4 2 4	2

Sample Input 2	**Sample Output 2**
6 11 1 2 1 3 1 5 2 3 2 4 2 6 3 4 3 5 3 6 4 5 5 6	4

Sample Input 3	**Sample Output 3**
252525 0	0

D. Distinct Numbers

Problem ID: distinct

You are given an array a of length n. For each k from 2 to n, find the maximum distance between two distinct elements within any subarray of length k.

Formally, for each k, we want to find:

$$\max\{j - i \mid 1 \le i < j \le n,\ j - i < k,\ a_i \ne a_j\}$$

If no such pair of distinct elements exists for a given k, the answer is -1.

Input

The first line contains a single integer n, representing the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$, representing the elements of the array.

Constraints

- $2 \le n \le 3 \times 10^5$
- $1 \le a_i \le 10^9$

Output

Output $n - 1$ space-separated integers on a single line. The i-th integer should be the answer for $k = i + 1$.

Sample Input 1	Sample Output 1
4 1 2 1 2	1 1 3

Sample Input 2

```
5
3 3 3 3 3
```

Sample Output 2

```
-1 -1 -1 -1
```

Sample Input 3

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

Sample Output 3

```
1 2 3 4
```

E. Electric Vehicle Revisited

Problem ID: ev2

This story is a work of fiction. Any similarity to real persons or events is coincidental.

After competing in the regional contest of the Fruit Kingdom, the team `vegetable_advantages` officially concluded their ICPC journey. However, this was also a new beginning because they wanted to give back to society: whenever they saw a problem, they would try to make it harder. While waiting for their flight back from the Fruit Kingdom to the Vegetable Kingdom, they happened to see this problem (which originated from 2025 National High School Programming Contest Problem C):

> A-Ming bought an electric vehicle with a battery capacity of B. A-Ming knows the initial battery level b, and he wants to plan a route from the starting point s to the destination t such that the total charging cost is as low as possible.
>
> The electric vehicle consumes battery power on certain road segments (such as flat roads or uphill roads) and charges the battery on other road segments (such as downhill roads). These charging road segments are **completely free of charge**. We represent the map using a directed graph, where the weight of an edge indicates the increase or decrease in battery level when driving across this edge. If driving through this edge charges the battery, the weight is a positive integer; conversely, if it consumes power, the weight is a negative integer. We assume that the graph has no positive cycles.
>
> While driving, the electric vehicle's battery level must always remain greater than or equal to 0, and no matter how much it charges, the battery level can at most reach B. More specifically, let p be the current battery level of the electric vehicle, and consider an edge with weight w: if w is non-negative, the electric vehicle can definitely drive through this edge (even if the battery level $p = 0$), and the remaining battery level will be $\min(B, p+w)$; if w is negative and $p + w \geq 0$, the electric vehicle can drive through this edge, and the remaining battery level after passing through will be $p + w$; however, if $p + w < 0$, the electric vehicle cannot drive through this edge.
>
> Some vertices on the map are charging stations. A-Ming can pass through multiple charging stations. Because charging takes time to find a charging pile, A-Ming decided to use **at most one charging station to charge** during the entire trip. Charging one unit of battery at a charging station costs one dollar. A-Ming's goal is to reach the destination while spending the least amount of money.

Cabbage, a member of `vegetable_advantages`, felt that A-Ming was far too lazy for being willing to use only one charging station! Besides, isn't A-Ming curious about the answers from the fixed starting point s to **every possible destination** $t = 1, 2, \ldots, n$? Therefore, on the plane, they solved the version where **one can charge at any number of charging stations** and must output the answers for $t = 1, 2, \ldots, n$, and decided to use it to test you.

Input

The first line contains three integers n, m, s, representing the number of vertices in the map, the number of edges, and the vertex index of the starting point, respectively.

The second line contains two integers B, b, representing the battery capacity and the initial battery level, respectively.

The following m lines each contain three integers u_i, v_i, w_i, representing that the i-th edge goes from vertex u_i to vertex v_i with weight w_i.

The first integer of the last line is g, representing the number of charging stations. This is followed by g integers $p_1, p_2, \ldots, p_g$, representing the vertex indices where the charging stations are located.

- $1 \le n \le 2\,000$
- $1 \le m \le 10^4$
- $1 \le s \le n$
- $1 \le B \le 10^9$
- $0 \le b \le B$
- $1 \le u_i, v_i \le n$, and $u_i \ne v_i$
- $-10^9 \le w_i \le 10^9$
- $0 \le g \le n$
- $1 \le p_1 < p_2 < \cdots < p_g \le n$
- It is guaranteed that there is no positive cycle in the graph.

Output

Output n integers, where the i-th integer represents the minimum amount of money A-Ming must spend to reach the destination when he can use any number of charging stations and $t = i$. If it is impossible to reach t, output -1.

Sample Input 1

```
7 7 1
100 20
1 2 -10
2 3 -5
3 4 -20
3 5 -30
4 6 -40
5 6 -10
6 7 20
1 3
```

Sample Output 1

```
0
0
0
15
25
35
35
```

Sample Input 2

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

Sample Output 2

```
0
1
2
3
4
```

Sample Input 3

```
4 3 2
1 0
2 1 0
1 3 -1
4 1 1
0
```

Sample Output 3

```
0
0
-1
-1
```

Note

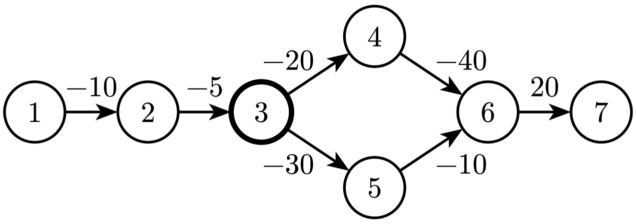
In Sample 1, A-Ming is initially at vertex $s = 1$ with an initial battery level of $b = 20$.

For each $t \in \{1, 2, 3\}$, there is only one path from s to t, and the initial battery level is sufficient for A-Ming to complete the journey without charging.

For each $t \in \{4, 5\}$, there is also only one path from s to t, but the initial battery level is insufficient. Thus, A-Ming has to charge his battery at vertex 3, which costs 15 dollars for $t = 4$ and 25 dollars for $t = 5$.

For $t = 6$, A-Ming takes the path $1 \to 2 \to 3 \to 5 \to 6$. He charges at vertex 3 up to a battery level of 40 so that he can traverse the edges $(3, 5)$ and $(5, 6)$. This costs 35 dollars.

For $t = 7$, A-Ming takes the path $1 \to 2 \to 3 \to 5 \to 6 \to 7$ and spends 35 dollars at vertex 3.



F. Flos[1] [2]

Problem ID: grid

Flowers may eventually wither, but the shapes they once formed can still be preserved on paper.

A botanist wishes to recreate a garden with as many flowers as possible on an $n \times m$ grid. Every cell of the grid must belong to exactly one flower. To represent a garden containing k flowers, each cell is assigned an integer from 1 to k, where a cell labeled i belongs to the i-th flower.

Each of the k flowers must occupy **at least two cells**. Moreover, for each flower, all cells belonging to it must be 4-connected. In other words, one must be able to travel between any two cells of the flower by moving only between side-adjacent cells belonging to the same flower.

The flowers in this garden must also be intertwined so that the grid cannot be separated into two parts by a single horizontal or vertical cut along grid lines. More precisely:

- For every pair of consecutive rows, there must be at least one flower that occupies cells both above and below the line between them.
- For every pair of consecutive columns, there must be at least one flower that occupies cells both to the left and to the right of the line between them.

Find the maximum possible number of flowers and construct a corresponding grid.

Input

The first line contains an integer t, denoting the number of test cases.

Each of the next t lines contains two integers n and m, denoting the number of rows and columns in that test case, respectively.

- $1 \leq t \leq 10\,000$
- $1 \leq n, m \leq 500$
- $n \times m \geq 2$
- The sum of $n \times m$ over all test cases does not exceed 500^2.

Output

For each test case, first output an integer k, the maximum possible number of flowers.

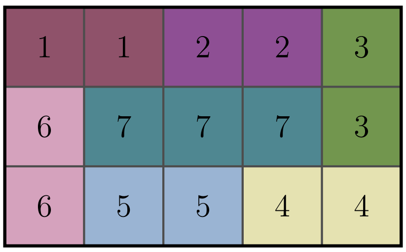
Then, output n lines, each containing m integers. The integer in row r and column c must be the label of the flower occupying that cell. Every label from 1 to k must appear, and the resulting grid must satisfy all the conditions described above.

If there are multiple valid grids, output any one of them.

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

Note

The following figure illustrates the output for the third test case in the sample. Cells with the same color and label belong to the same flower.



For example, flower 7 occupies three consecutive cells in the second row. The figure also shows that every horizontal or vertical line between consecutive rows or columns is crossed by at least one flower. It can be shown that no valid garden can contain 8 or more flowers.

G. Great Cactus Comeback

Problem ID: cactus

Did you know that cactus graphs once enjoyed a glorious era in Taiwan's competitive programming scene, especially in 2022?

Looking back, the 2021 NTU ICPC Team Preliminary Contest featured a problem that required answering multiple shortest-path distance queries on a cactus graph[1], marking the beginning of the Great Cactus Era. In 2022, cactus graphs reached the height of their glory. First, IOICamp featured a problem about answering maximum contiguous path-sum queries between pairs of vertices in a cactus graph[2]. Later, YTP featured a problem asking for the maximum-weight independent set of a cactus graph[3]. Incidentally, this problem was so difficult that even the first-place team solved every problem except this one! Finally, that year's NTU ICPC Team Preliminary Contest featured yet another problem combining cactus graphs with linear basis[4].

Unfortunately, since 2023, cactus graphs have almost completely disappeared from programming contests in Taiwan. What a shame! Now, you are given a simple, connected, undirected cactus graph with n vertices and m edges. Each edge has a positive integer weight. There is a set of vertices S which is initially empty. Please process q operations. Each operation specifies a vertex v and toggles whether v belongs to S. That is,

- If $v \in S$, remove v from S;
- If $v \notin S$, add v to S.

After each operation, output the maximum shortest-path distance between any two vertices in S. That is, let $d(u, v)$ be the length of the shortest path between vertices u and v. You should output

$$\max_{u,v \in S} d(u, v)$$

after each operation. Note that if $|S| \leq 1$, the answer is defined to be 0.

To make the problem more interesting, all operations are encoded and must be processed online. See the input format for details.

Note that an undirected graph is called a cactus graph if every edge belongs to at most one simple cycle.

Input

The first line contains three integers n, m, and q, denoting the number of vertices, the number of edges, and the number of operations, respectively.

Each of the next m lines contains three positive integers u_i, v_i, and w_i, indicating that there is an undirected edge of weight w_i between vertices u_i and v_i.

Each of the next q lines contains an integer x_i, representing the encoded information for the i-th operation.

Let ans_i be the answer after the i-th operation, and let y_i be the vertex that is actually toggled by the i-th operation. Then

$$y_i = \text{ans}_{i-1} \oplus x_i,$$

where $\oplus$ denotes the bitwise XOR operation. Also, we define $\text{ans}_0 = 0$.

- $1 \leq n, q \leq 10^5$
- $0 \leq m \leq 2 \times 10^5$
- $1 \leq u_i, v_i \leq n$
- $1 \leq w_i \leq 10^9$
- $0 \leq x_i < 2^{60}$
- $1 \leq y_i \leq n$
- It is guaranteed that the input graph is a simple, connected cactus graph.

Output

Output q lines. The i-th line should contain one integer ans_i, the maximum shortest-path distance between any two vertices in S after the i-th operation.

Sample Input 1

```
15 19 20
10 13 4
11 1 2
14 12 5
9 5 5
8 5 9
14 8 10
15 11 5
13 3 3
6 3 6
13 6 10
13 7 1
1 15 5
4 15 8
7 10 8
14 2 6
14 10 5
12 8 1
4 3 7
2 10 7
6
14
31
27
36
44
37
42
47
17
30
44
18
13
0
41
6
8
31
30
```

Sample Output 1

```
0
18
18
38
38
38
38
38
23
17
32
29
14
11
34
11
7
29
27
42
```

Sample Input 2

```
8 8 10
2 1 5
2 7 3
8 5 4
5 4 1
6 4 5
3 2 4
1 6 5
4 2 2
2
8
2
6
13
4
9
15
10
4
```

Sample Output 2

```
0
7
7
12
7
11
11
11
12
9
```

Note

In the first sample, the actual query are: 2 8 9 12 3 5 13 6 6 9 6 6 15 14 15 4 4 5 10 1.

In the second sample, the actual query are: 2 8 5 1 1 3 2 4 1 8.

H. Hexagon Game

Problem ID: deque

If you ask Doludu and DoIudu what their favorite shape is, they will definitely say it is a hexagon. Because they really love hexagons so much, no matter what object it is, they always want to design it in a hexagonal shape, and game items are no exception.

Today, they are going to play a game. In this game, there are N hexagonal blocks lined up in a row at the beginning, and these hexagonal blocks are divided into blue and red. After the game starts, the two of them will take turns to make a move, with Doludu going first. When it is a player's turn, he must remove either the leftmost or the rightmost block among the currently remaining blocks. At the moment when only one block is left, the game ends. If this block is blue, Doludu wins; otherwise, DoIudu wins.

Both Doludu and DoIudu are extremely smart and will definitely use the optimal strategy to play this game. Based on the initial colors of the N blocks, please determine who will win among them.

Input

The first line of the input contains an integer T, representing that Doludu and DoIudu will play T games.

Next are the details of each game in order. The information for a single game consists of two lines. The first line contains an integer N, representing the number of hexagonal blocks. The second line contains a string S of length N, where the i-th character S_i represents the color of the i-th block from the left, with B representing blue and R representing red.

- $1 \leq T \leq 10^5$
- $2 \leq N \leq 2 \times 10^5$
- The sum of N over all games does not exceed 2×10^5.
- $S_i \in \{\mathsf{B}, \mathsf{R}\}$

Output

Output T lines, where the i-th line outputs the result of the i-th game. If Doludu wins, output `Doludu`; otherwise, output `DoIudu`.

Sample Input 1

```
4
5
RRRRR
4
RBRB
4
BRRB
20
RBRBRBBRBRBBRRBRBRBR
```

Sample Output 1

```
DoIudu
Doludu
DoIudu
Doludu
```

I. Inversion Counting

Problem ID: inversion

Given a permutation a from 1 to n, an inversion is defined as a pair of indices (i, j) such that $1 \le i < j \le n$ and $a_i > a_j$.

Counting the total number of inversions is an extremely classic problem. In this problem, we consider a variation.

For each k from 1 to $n - 1$, find the number of pairs (i, j) such that $1 \le i < j \le n$, $j - i = k$, and $a_i > a_j$.

Input

The first line contains a single integer n, representing the length of the sequence.

The second line contains n integers $a_1, a_2, \ldots, a_n$, representing the permutation.

Constraints

- $2 \le n \le 2.5 \times 10^5$
- $a_1, a_2, \ldots, a_n$ is a permutation from 1 to n.

Output

Output $n - 1$ space-separated integers on a single line. The k-th integer should represent the number of pairs (i, j) satisfying the conditions for the given value of k.

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

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

This page is intentionally left blank.

J. Jester's Paint Parade

Problem ID: jester

In the Grand Palace of the Fruits Kingdom, there is a long ornamental chain consisting of n decorative beads, numbered from 1 to n from left to right. Initially, all beads are uncolored.

To celebrate the upcoming Fruits' Festival, the King has invited m jesters to color the beads. The jesters are numbered from 1 to m. The coloring process runs for m turns, in the i-th turn, the i-th jester must paint a **contiguous** segment of the beads with color i. However, the jesters do not view themselves as ordinary performers, instead, all of them are strictly adhered to the sacred number k, insisting that each jester's stroke must cover **exactly** k **beads**. It follows that each jester has $n - k + 1$ choices in their turn.

Note that painting a colored bead will overwrite its original color.

Since the jesters are paid, the king does not want to see uncolored beads at the ceremony. Thus, all n beads must be painted by at least one of the jesters by the end of the process.

Please help the King calculate the total number of possible final states of the beads that satisfy the condition above. Two final states are considered different if and only if there exists a bead i that has a different color in the two states. Since the answer could be large, output it modulo p.

Input

The input contains four integers n, m, k, p, representing the number of beads, the number of jesters, the sacred number and the modulo.

- $2 \leq k \leq n \leq 4000$
- $1 \leq m \leq 10^8$
- $10^8 \leq p \leq 10^9$, p is prime.

Output

Output a single integer, representing the total number of valid final states modulo p.

Sample Input 1	Sample Output 1
`4 2 3 588100627`	`2`

Sample Input 2	Sample Output 2
`5 3 2 573793837`	`10`

Sample Input 3	Sample Output 3
`4000 100000000 30 809157511`	`808328768`

Notes

In the first sample, there are two possible coloring processes:

- The first jester paints beads $1, 2, 3$, and the second jester paints beads $2, 3, 4$. The final state is $[1, 2, 2, 2]$.
- The first jester paints beads $2, 3, 4$, and the second jester paints beads $1, 2, 3$. The final state is $[2, 2, 2, 1]$.

Note that they can't both paint beads $1, 2, 3$, since bead 4 will be uncolored and violate the King's requirement.

K. Kuusi Seitsemän

Problem ID: sixseven

Six seven. Six seven. 67, 67, 67.

"It's joever; pack it up," you thought. The brainrot trend has gone too far. It is the end of society.

First they came as skibidi toilet, and you did not speak out. Because you thought that was fine. Then they came as tralalero tralala, and you did not speak out. Because you thought that was nonsense. Then they released the game Tung Tung Sahur Extreme, and you did not speak out. Because you thought no one cared about that. Then they came as six seven, and there was no one left to speak for you. The problem title itself, you realized, it means "six seven" in Finnish. You are finished.

All that is left, is a string. You are given a string S. It can contain three different characters: six (`6`), seven (`7`), and — if you think that's enough, it is not — ASCII code 67: upper case Latin alphabet `C`.

This is your final chance. Even if it might be futile, you would still want to try for the very last time. You must replace each ASCII code 67 character with one of six (`6`) or seven (`7`). Your goal is to minimize the number of subsequences of the string that are identical to six seven (`67`).

Two subsequences are considered different if there exists at least one position that is in one but not the other.

Input

The first line contains a single integer t, indicating the number of test cases. Next, t lines follows. Each test case consists of single line containing the string S.

Constraints

- $1 \leq t \leq 10^5$
- $1 \leq |S|$
- The sum of the lengths of all strings S does not exceed 5×10^5.
- Each character of S is either `6` or `7`, or has ASCII code 67.

Output

For each test case, output one line containing an integer, indicating the minimum number of subsequences that are identical to `67` after substituting all character with ASCII code 67.

Sample Input 1	Sample Output 1
5 67CCC67 67C7CC67 6666C6767CC6C6C67777 6676667667767 7C67C77 6666666666766766C6666666C6677C77776777777C77677777777777	6 7 67 69 727

L. Largest Prime Factor

Problem ID: lpf

This is Mango's dark history.

One day, he learned about the linear sieve, an algorithm capable of computing multiplicative functions, such as $\phi(n)$ and $\mu(n)$, in linear time. However, when explaining this algorithm to his teammates, he made a silly mistake. He confused the abbreviation LPF with Largest Prime Factor, as it is quite natural for a non-native English speaker to mistake least for largest.

To ensure he never forgets this embarrassing moment, he brings this problem to you. Given an integer r, calculate

$$\sum_{i=1}^{r} G(i)$$

where $G(i)$ denotes the largest prime factor of i. He uses the letter G because it is more common to call it Greatest Prime Factor.

For convenience, define $G(1) = 0$.

Input

The only line of input contains a single integer r.

- $1 \leq r \leq 10^{10}$

Output

Output a single integer representing the value of $\sum_{i=1}^{r} G(i)$.

It is guaranteed that the answer fits within a standard signed 64-bit integer.

Sample Input 1	Sample Output 1
5	12

Sample Input 2	Sample Output 2
11	43

Sample Input 3	Sample Output 3
10000000000	3755757137823525252

Notes

In the first sample, $\sum_{i=1}^{5} = G(1) + G(2) + G(3) + G(4) + G(5) = 0 + 2 + 3 + 2 + 5 = 12$.

In the second sample, $\sum_{i=1}^{11} = 0 + 2 + 3 + 2 + 5 + 3 + 7 + 2 + 3 + 5 + 11 = 43$.

M. Merging Slimes

Problem ID: merge

In a certain RPG, there are n slimes lined up in a row, numbered from 1 to n from left to right. Each slime can be of water-type, fire-type, or elementless. The power of the i-th slime is represented by a signed integer a_i. Specifically:

- If $a_i > 0$, the slime is of water-type, and its power is a_i.
- If $a_i < 0$, the slime is of fire-type, and its power is $|a_i|$ (which is equal to $-a_i$).
- If $a_i = 0$, the slime is elementless, and its power is 0 (representing no energy or power value).

As a player, you will use a **water-type** magic to attack these slimes. The mana cost to attack a single slime is determined as follows:

- If its represented power $a_i > 0$, the mana cost is a_i.
- If its represented power $a_i \leq 0$ (meaning the slime is fire-type or elementless), the mana cost is 0.

To minimize the total mana cost, you can perform at most k merging operations before casting the magic:

- Select two adjacent slimes with represented powers a and b, and merge them into a single slime. The represented power of the newly merged slime will be $a + b$.

Note that if two slimes of opposite types and equal power are merged, they will neutralize each other, resulting in an elementless slime with a power of 0.

After performing at most k operations, you cast the magic. The total final cost will be the sum of the mana costs required to attack all remaining slimes.

Find the minimum possible total cost if you use an optimal strategy.

Input

The first line contains two integers n and k, representing the number of slimes and the maximum number of merging operations you can perform.

The second line contains n integers $a_1, a_2, \ldots, a_n$, representing the initial represented powers of the slimes.

Constraints

- $1 \leq n \leq 3000$
- $0 \leq k < n$
- $-10^9 \leq a_i \leq 10^9$

Output

Output a single integer, representing the minimum possible total cost.

Sample Input 1	Sample Output 1
3 1 10 -10 5	5

Sample Input 2	Sample Output 2
5 2 3 -5 4 -2 6	7

Sample Input 3	Sample Output 3
4 2 5 -5 8 -8	0

Notes

In the first sample, you can merge the first two slimes, a water-type slime of power 10 and a fire-type slime of power 10. They neutralize each other and become an elementless slime, so the row becomes $[0, 5]$. Attacking the elementless slime costs 0 mana and attacking the remaining water-type slime of power 5 costs 5 mana, for a total cost of $0 + 5 = 5$.

In the second sample, you can use your two operations on two disjoint pairs of adjacent slimes: merge the second and the third slimes into a fire-type slime of power 1 (since $-5 + 4 = -1$), and merge the fourth and the fifth slimes into a water-type slime of power 4 (since $-2 + 6 = 4$). The row becomes $[3, -1, 4]$, and the total cost is $3 + 0 + 4 = 7$. No strategy using at most 2 operations can do better.

In the third sample, you can merge the first two slimes and then the last two slimes. Each pair consists of slimes of opposite types with equal power, so both merges produce elementless slimes and the row becomes $[0, 0]$. Attacking them costs no mana at all, so the total cost is 0.

This page is intentionally left blank.