Topic 4: Monotonicity: DP Review, Optimization — **Handout**
CS 41100 - CP3 Competitive Programming III (Spring 2024) — **Purdue University**
Instructor: Zhongtang Luo — **Date:** February 5, 2024

Sample Problem: B Road Band
Link: `https://vjudge.net/problem/Gym-104757B`

Sample Problem: Watching Fireworks is Fun
Link: `https://vjudge.net/problem/CodeForces-372C`

Sample Problem: Covered Walkway
Link: `https://vjudge.net.cn/problem/Kattis-coveredwalkway`

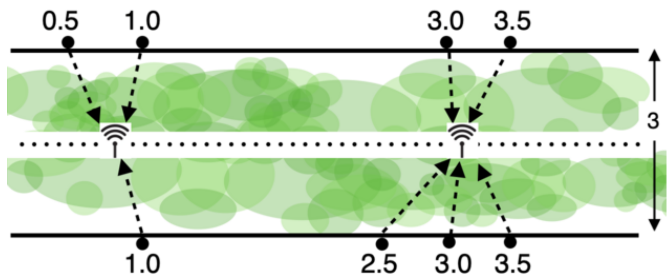
Sample Problem: The Elder
Link: `https://vjudge.net.cn/problem/HDU-5956`

Sample Problem: The Bakery
Link: `https://vjudge.net/problem/CodeForces-834D`

Sample Problem: Monkey Party
Link: `https://vjudge.net/problem/HDU-3506`

B Road Band

All the residents of the rural community of Axes Point live on one of two parallel streets separated by a band of green park land. Recently, the local board of supervisors received a grant to (finally) bring wireless service to the town. The grant provides enough money for them to install k access points, and the supervisors have decided to place them in a straight line on County Road "B," which lies in the wooded band midway between the two residential streets. They want to place them in a way that minimizes the distance between users and their nearest access point. Specifically, they want to minimize the sum of the squares of the distances of each user from their nearest access point. For instance, Figure 1 shows two streets with eight customers and their locations along the streets (this is the first sample input). The streets are 3 units apart, and two access points have been placed at points midway between the two streets so that the sum of the eight squared distances is minimized.



Sample Input 1 showing placement of access points

Given the locations of all customers along each of the two streets, the distance between the streets, and the number of access points, help the local government determine the minimum sum of squared distances that can be achieved.

Input

There are three lines of input. The first line contains four integers m, n, k, s, where m and n ($1 \leq m, n \leq 1\,000$) are the number of customers along each of the two roads, k ($1 \leq k \leq \min(\max(m, n), 100)$) is the number of access points to be placed, and s ($1 \leq s \leq 50$) is the distance separating the two roads. The second line contains m floating-point values $x_1, x_2, \cdots, x_m$ ($0 \leq x_i \leq 1\,000$) giving the locations of the m customers along the first road. The third line is similar, containing n floating-point locations of customers along the second road. All values on each of the second and third lines will be distinct (but some values may appear in both lines). Customer locations will have no more than four decimal places.

Output

Output a single floating-point value equal to the minimum sum of squared distances for each customer from the closest of the k access points. Answers should be correct to within an absolute or relative error of 10^{-5}.

Examples

Input

```
4 4 2 3
0.5 1.0 3.0 3.5
1.0 2.5 3.0 3.5
```

Output

```
18.86666667
```

Source

2023-2024 ICPC East North America Regional Contest (ECNA 2023)

Watching Fireworks is Fun

A festival will be held in a town's main street. There are n sections in the main street. The sections are numbered 1 through n from left to right. The distance between each adjacent sections is 1.

In the festival m fireworks will be launched. The i-th ($1 \le i \le m$) launching is on time t_i at section a_i. If you are at section x ($1 \le x \le n$) at the time of i-th launching, you'll gain happiness value $b_i - |a_i - x|$ (note that the happiness value might be a negative value).

You can move up to d length units in a unit time interval, but it's prohibited to go out of the main street. Also you can be in an arbitrary section at initial time moment (time equals to 1), and want to maximize the sum of happiness that can be gained from watching fireworks. Find the maximum total happiness.

Note that two or more fireworks can be launched at the same time.

Input

The first line contains three integers n, m, d ($1 \le n \le 150000$; $1 \le m \le 300$; $1 \le d \le n$).

Each of the next m lines contains integers a_i, b_i, t_i ($1 \le a_i \le n$; $1 \le b_i \le 10^9$; $1 \le t_i \le 10^9$). The i-th line contains description of the i-th launching.

It is guaranteed that the condition $t_i \le t_{i+1}$ ($1 \le i < m$) will be satisfied.

Output

Print a single integer — the maximum sum of happiness that you can gain from watching all the fireworks.

Examples

Input

```
50 3 1
49 1 1
26 1 4
6 1 10
```

Output

```
-31
```

Input

```
10 2 1
1 1000 4
9 1000 4
```

Output

```
1992
```

Source

Codeforces Round 219 (Div. 1)

Covered Walkway

Your university wants to build a new walkway, and they want at least part of it to be covered. There are certain points which must be covered. It doesn't matter if other points along the walkway are covered or not.

The building contractor has an interesting pricing scheme. To cover the walkway from a point at x to a point at y, they will charge $c + (x - y)^2$, where c is a constant. Note that it is possible for $x = y$. If so, then the contractor would simply charge c.

Given the points along the walkway and the constant c, what is the minimum cost to cover the walkway?

Input

Input consists of a single test case. The test case will begin with a line with two integers, n ($1 \leq n \leq 1\,000\,000$) and c ($1 \leq c \leq 10^9$), where n is the number of points which must be covered, and c is the contractor's constant. Each of the following n lines will contain a single integer, representing a point along the walkway that must be covered. The points will be in order, from smallest to largest. All of the points will be in the range from 1 to 10^9, inclusive.

Output

Output a single integer, representing the minimum cost to cover all of the specified points. All possible inputs yield answers which will fit in a signed 64-bit integer.

Examples

Input

```
10 5000
1
23
45
67
101
124
560
789
990
1019
```

Output

```
30726
```

Source

2012 University of Chicago Invitational Programming Contest

The Elder

Once upon a time, in the mystical continent, there is a frog kingdom, ruled by the oldest frog, the Elder. The kingdom consists of N cities, numbered from east to west. The 1-st city, which is located to the east of others, is the capital. Each city, except the capital, links none or several cities to the west, and exactly one city to the east.

There are some significant news happening in some cities every day. The Elder wants to know them as soon as possible. So, that is the job of journalist frogs, who run faster than any other frog. Once some tremendous news happen in a city, the journalist in that city would take the message and run to the capital. Once it reach another city, it can either continue running, or stop at that city and let another journalist to transport. The journalist frogs are too young and simple to run a long distance efficiently. As a result, it takes L^2 time for them to run through a path of length L. In addition, passing message requires P time for checking the message carefully, because any mistake in the message would make the Elder become extremely angry.

Now you are excited to receive the task to calculate the maximum time of sending a message from one of these cities to the capital.

Input

The first line of input contains an integer t, the number of test cases. t test cases follow. For each test case, in the first line there are two integers N ($N \leq 100000$) and P ($P \leq 1000000$). In the next $N - 1$ lines, the i-th line describes the i-th road, a line with three integers u,v,w denotes an edge between the u-th city and v-th city with length w ($w \leq 100$).

Output

For each case, output the maximum time.

Examples

Input

```
3
6 10
1 2 4
2 3 5
1 4 3
4 5 3
5 6 3
6 30
1 2 4
2 3 5
1 4 3
4 5 3
5 6 3
6 50
1 2 4
2 3 5
1 4 3
4 5 3
5 6 3
```

Output

```
51
75
```

```
81
```

Note

In the second case, the best transportation time is:

- The 2-th city: $16 = 4^2$
- The 3-th city: $72 = 4^2 + 30 + 5^2$
- The 4-th city: $9 = 3^2$
- The 5-th city: $36 = (3 + 3)^2$
- The 6-th city: $75 = (3 + 3)^2 + 30 + 3^2$

Consequently, the news in the 6-th city requires most time to reach the capital.

Source

2016 ACM/ICPC Shenyang Regional

The Bakery

Some time ago Slastyona the Sweetmaid decided to open her own bakery! She bought required ingredients and a wonder-oven which can bake several types of cakes, and opened the bakery.

Soon the expenses started to overcome the income, so Slastyona decided to study the sweets market. She learned it's profitable to pack cakes in boxes, and that the more **distinct** cake types a box contains (let's denote this number as the *value* of the box), the higher price it has.

She needs to change the production technology! The problem is that the oven chooses the cake types on its own and Slastyona can't affect it. However, she knows the types and order of n cakes the oven is going to bake today. Slastyona has to pack exactly k boxes with cakes today, and she has to put in each box several (at least one) cakes the oven produced one **right after another** (in other words, she has to put in a box a continuous segment of cakes).

Slastyona wants to maximize the total value of all boxes with cakes. Help her determine this maximum possible total value.

Input

The first line contains two integers n and k ($1 \le n \le 35000$, $1 \le k \le min(n, 50)$) – the number of cakes and the number of boxes, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$) – the types of cakes in the order the oven bakes them.

Output

Print the only integer – the maximum total value of all boxes with cakes.

Examples

Input

```
4 1
1 2 2 1
```

Output

```
2
```

Input

```
7 2
1 3 3 1 4 4 4
```

Output

```
5
```

Input

```
8 3
7 7 8 7 7 8 1 7
```

Output

```
6
```

Note

In the first example Slastyona has only one box. She has to put all cakes in it, so that there are two types of cakes in the box, so the value is equal to 2.

In the second example it is profitable to put the first two cakes in the first box, and all the rest in the second. There are two distinct types in the first box, and three in the second box then, so the total value is 5.

Monkey Party

Far away from our world, there is a banana forest. And many lovely monkeys live there. One day, SDH(Song Da Hou), who is the king of banana forest, decides to hold a big party to celebrate Crazy Bananas Day. But the little monkeys don't know each other, so as the king, SDH must do something.

Now there are n monkeys sitting in a circle, and each monkey has a making friends time. Also, each monkey has two neighbor. SDH wants to introduce them to each other, and the rules are:

1. every time, he can only introduce one monkey and one of this monkey's neighbor.
2. if he introduce A and B, then every monkey A already knows will know every monkey B already knows, and the total time for this introducing is the sum of the making friends time of all the monkeys A and B already knows;
3. each little monkey knows himself;

In order to begin the party and eat bananas as soon as possible, SDH want to know the mininal time he needs on introducing.

Input

There is several test cases. In each case, the first line is n $(1 \leq n \leq 1000)$, which is the number of monkeys. The next line contains n positive integers (less than 1000), means the making friends time (in order, the first one and the last one are neighbors). The input is end of file.

Output

For each case, you should print a line giving the mininal time SDH needs on introducing.

Examples

Input

```
8
5 2 4 7 6 1 3 9
```

Output

```
105
```

Note

2010 ACM-ICPC Multi-University Training Contest 7 — Host by HIT