TASK	GLASNICI	KOLEKCIJA	TAMNICA	UMNOZAK
input	standard input			
output	standard output			
time limit	1 second	1.5 seconds	1 second	1 second
memory limit	64 MB			
points	**100**	**100**	**100**	**100**
	400			

A long straight road connects two villages. Along the road, N messengers are stationed and, when needed, they exchange messages using mostly their legs, but also their vocal cords and ears.

The first messenger (the closest to the first village) has a radio-receiver which he uses to keep track of current ongoings in the country. When he finds out who has been evicted from whichever reality show is currently popular, he starts running as fast as he can to share the unfortunate (or fortunate) news with everyone else. While running, he shouts the name of the evicted person so that any fellow messengers that are close enough can hear him. Meanwhile, the remaining messengers do not merely sit and wait, but also run themselves, all with the selfless goal of sharing the news with everyone as fast as possible.

The running and shouting proceeds as follows:

- Each of the messengers may run **whenever**, **in either direction**, at a speed of **at most 1 unit per second**, or may decide not to run at all and stand still.
- All messengers that know the news shout it at all times. One messenger **can hear** another messenger shouting (and learn the news) if the **distance** between them is **at most K units**.

Write a program that, given the initial locations of the messengers, determines the **least amount of time** (in seconds) needed for **all messengers** to learn the news. The location of every messenger is given with a positive real number – the distance from the first village. As mentioned above, initially only the first messenger knows the news.

INPUT

The first line contains the real number K ($0 \leq K \leq 10^6$), the largest distance at which two messengers can hear each other.

The second line contains the integer N ($1 \leq N \leq 100000$), the number of messengers.

Each of the following N lines contains one real number D ($0 \leq D \leq 10^9$), the distance of one messenger from the first village. The distances will be sorted **in ascending order**. It is possible for multiple messengers to be at the same location.

OUTPUT

Output a real number, the least time for all messengers to learn the news. Your output will be accepted if it differs from the official output by no more than ± 0.001.

EXAMPLE TEST DATA

input

```
3.000
2
0.000
6.000
```

output

```
1.500
```

input

```
2.000
4
0.000
4.000
4.000
8.000
```

output

```
1.000
```

Igor has a huge collection of folk hits on his computer, containing N songs numbered 1 to N.

The collection is so big that it is not possible to display all songs at once on his display. Because of this, while a song is playing, only **K consecutive songs** from the collectionare displayed on the screen. Of course, the K consecutive songs necessarily **include the song currently playing**.

When a song first appears on the display, the software needs to access its file on disk and read metadata like artist and song name. This metadata is stored in the computer's memory so that, if the song reappears on display, the file doesn't need to be opened again.

Your program will be given the songs Igor wants to listen to, in the order in which he wants to do it. For each song, determine the interval of songs which will be displayed while it is playing, so that the **total number of files** that need to be accessed on disk is the **smallest possible**.

Note: The solution may not be unique.

INPUT

The first line contains two integers N and K ($1 \leq K < N < 1\,000\,000\,000$), the number of songs in the collection and the number of songs displayed.

The second line contains the integer M ($1 \leq M \leq 300\,000$), the number of songs Igor will listen to.

The next M lines contain the indices of the songs Igor will listen to. All numbers will be between 1 and N and no song will appear more than once.

OUTPUT

Output should consist of M+1 lines.

On the first line output the **smallest possible** number of files to access while playing Igor's playlist.

After this, for each song S, in order in which they are given, output a pair of integers A and B, meaning that while song S is playing, songs A through B (inclusive) are displayed on screen. A and B must satisfy the conditions $1 \leq A \leq S \leq B \leq N$, and $B-A+1 = K$.

SCORING

An output which is not completely correct, but the first line (the least number of file to access) is correct, will score 50% points for that test case.

EXAMPLE TEST DATA

input

```
10 3
5
4
5
8
7
6
```

output

```
5
4 6
4 6
6 8
6 8
6 8
```

input

```
15 4
6
6
14
11
3
8
5
```

output

```
10
3 6
11 14
11 14
3 6
5 8
3 6
```

input

```
1000 301
3
300
500
700
```

output

```
401
300 600
350 650
400 700
```

Brave Sir Robin has been thrown in the dungeon by the evil king. The dungeon consists of an infinite number of cube-shaped rooms with big stone walls. Rooms are connected by passages so that the entire dungeon, when viewed from above, looks like a spiral. The rooms are numbered as follows:

...	35	34	33	32	31
17	16	15	14	13	30
18	5	4	3	12	29
19	6	1	2	11	28
20	7	8	9	10	27
21	22	23	24	25	26

After a big earthquake **some of the walls collapsed**, and new passages were formed between adjacent rooms.

Sir Robin is initially in room 1. Sir Robin knows that the exit from the dungeon is located in room N, and wants to escape while everyone is distracted by the earthquake. Because the evil dragon is guarding the dungeon, Sir Robin wants to use the fastest way out of the dungeon.

Write a program that, given the location of the exit N and the list of new passages, determines the **smallest number of passages** that Sir Robin must go through before he can exit the dungeon.

INPUT

The first line of input contains an integer N ($1 \leq N \leq 10^{15}$), the room in which the exit is located.

The second line of input contains an integer K ($1 \leq K \leq 100\,000$), the number of new passages.

Each of the following K lines contains one integer B ($4 \leq B \leq 10^{15}$), meaning that a new passage now connects adjacent rooms A and B, where A<B. The number A is not given explicitly, but it can be uniquely determined from B (for example, if B is 20, then A must be 7). Also, some rooms can never be room B (rooms 2, 3, 5, 7, 10, 13 etc.).

OUTPUT

Output should consist of a single integer, the smallest number of passages that Sir Robin must go through before he can exit the dungeon.

SCORING

In a number of test cases, worth a total of 50 points, N will be at most 10^6.

EXAMPLE TEST DATA

input

```
31
9
15
25
30
6
9
19
24
27
4
```

output

```
6
```

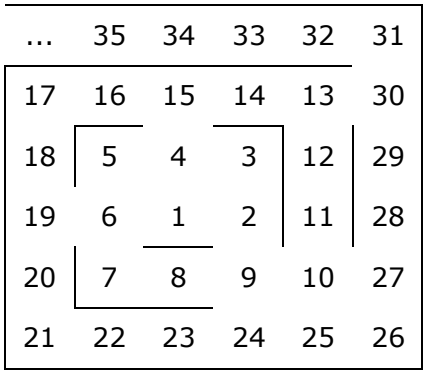
input

```
10000
5
52
4
9
25
27
```

output

```
9953
```

Clarification of first example. This is the layout of the dungeon after the earthquake:



Mirko can use the route 1-4-15-14-13-30-31, using only 6 hallways to exit the dungeon.

The **digit-product** of a positive integer is the product of the number's decimal digits. For example, the digit-product of 2612 is $2 \cdot 6 \cdot 1 \cdot 2 = 24$.

The **self-product** of a number is the product of the number and its digit-product. For example, the self-product of 2612 is $2612 \cdot 24 = 62688$.

Write a program that, given two positive integers A and B, calculates the number of positive integers whose self-product is between A and B, inclusive.

INPUT

The first and only line contains two integers A and B ($1 \leq A \leq B < 10^{18}$).

OUTPUT

Output should consist of a single integer, the number of positive integers whose twist is between A and B.

SCORING

In test cases worth a total of 25 points, A and B will be at most 10^8.

In test cases worth another 15, A and B will be at most 10^{12}.

EXAMPLE TEST DATA

input	input	input
20 30	145 192	2224222 2224222
output	**output**	**output**
2	4	1

Clarification of second example. The self-products of numbers 19, 24, 32 i 41 are in order 171, 192, 192 and 164.