

COI '07 #2 Kolekcija

Time limit: 0.75s **Memory limit:** 64M

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 collection are 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 Specification

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 Specification

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 files to access) is correct, will score 50% points for that test case.

Sample Input 1

10 3
5
4
5
8
7
6

Sample Output 1

5
4 6
4 6
6 8
6 8
6 8

Sample Input 2

15 4
6
6
14
11
3
8
5

Sample Output 2

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

Sample Input 3

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

Sample Output 3

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