Problem I. Integer Number Format

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

A new format IIII-4 of storing integer numbers is being developed. Encoding of each integer results in one, two, three, four or five 4-bit groups. First group is mandatory and then follow from zero to four optional additional groups.

Decoding is done as follows.

1. Read first 4-bit mandatory group.
2. Look it up in a table that contains the number of additional groups g and the offset f.
3. Read g additional groups. Decode this groups as an integer y from 0 to $2^{4g} - 1$.
4. Add offset to y, i.e. the result is $y + f$.

You are given two integers x and y and a sequence of integers $a_1, a_2, \ldots, a_n$, such that $x \le a_i \le y$. Your goal is to decide on a look up table and offsets in a way that allows to encode and decode all integers from x to y inclusive and the length of the encoding of the input sequence is minimum possible.

Note that the order of bits that is used to decode the number from 0 to $2^{4g} - 1$ doesn't matter for the purpose of this problem.

There should be a unique way to encode each integer from x to y using this table. However, integers outside of this range can be encoded in any number of ways.

Input

The first line contains two integers x and y ($-10^9 \le x \le y \le 10^9$, $y - x \le 1\,000\,000$) — bounds of the range that we need to be able to encode with the method.

The second line contains a single integer n ($1 \le n \le 100\,000$) — the length of the sequence to be encoded. The third line contains n integers $a_1, a_2, \ldots, a_n$ ($x \le a_i \le y$).

Output

Print the optimal table represented by 16 lines corresponding to decoding information of 16 prefixes.

Each line should contain two integers: g_i ($0 \le g_i \le 4$) stands for the number of additional groups and f_i ($-2 \cdot 10^9 \le f_i \le 2 \cdot 10^9$) is the offset.

The given sequence of numbers must be encoded to the minimum number of bits by the table. It is guaranteed that at least one valid encoding exists. If there are several optimal tables, you can print any of them.

Examples

standard input	standard output
0 15 16 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0 0 0 1 0 2 0 3 0 4 0 5 0 6 0 7 0 8 0 9 0 10 0 11 0 12 0 13 0 14 0 15
-128 127 4 1 0 -1 0	2 -261 0 -5 0 -4 0 -3 0 -2 0 -1 0 0 0 1 1 2 1 18 1 34 1 50 1 66 1 82 1 98 1 114