Parrots are intelligent. They can be trained to speak! Therefore, they can be used to carry messages over long distances.

The basic idea is to let a parrot remember the message and repeat it to the receiver (a microphone or something else that can record sound). Due to the limited memory of the parrot, this mechanism will fail when the message is long.

To handle a long message, a naive solution is that we divide the message into multiple words for transmission. However, there are two major problems for this mechanism:

- In the absence of context, words are difficult to identify.
- Parrots may arrive in arbitrary order, making it hard or even impossible to reconstruct the message.

The first problem can be solved by using specially designed words instead of English words in the transmission. In practice, we carefully choose k special words with the same length in pronunciation. Each parrot only transmit one of the k words instead of a part of the raw message.

The second problem can be solved by introducing special coding schemes. Fortunately, you do not need to worry about coding schemes because this problem is already solved by tourist in IOI2011 `:-)`.

Do we solve all the problems till now? No! As a result of applying such a special coding scheme, the encoded messages are no longer understandable by just listening to the sound. An automatic decoding system is essential. And guess what? The task of implement the decoding system is on you!

The receiver records a piece of waveform as a series of integers. Given the recorded waveform S which consists of n integers, the waveforms of the k special words $W_1 \dots W_k$ (each consists of m integers), and the number of words of the message (also the number of parrots) l, your decoding system should recover the l words transmitted by the parrots.

Parrots are well-behaved, so no two parrots speak at the same time. But there may be some periods of time without any parrot speaking, and during such periods, the receiver also record some useless data. So n may be larger than $l * m$.

Since the sound we receive may be distorted for a variety of reasons, you only need to recover the words that are most likely to be the original ones. To be more specific, we define *Decoding Result* and *Similarity Value*. A *Decoding Result* is that you choose l none-overlapping sub-waveforms (a sub-waveform is a consecutive part of S) of length m from S and match each of them with one of the k special words. The *Similarity Value* of a sub-waveform A to its matched word B is defined as $\sum_{i=0}^{m-1}(a_i - b_i)^2$ (A is $a_0, a_1 \dots a_{m-1}$ and B is $b_0, b_1 \dots b_{m-1}$). The *Similarity Value* of a *Decoding Result* is defined as the sum of the *Similarity Values* of all the l chosen sub-waveforms to their matched words.

Your task is to find a *Decoding Result* with the minimum *Similarity Value*.

Input

The first line contains a single integer T, indicating the number of test cases.

In each test case, the first line contains 4 integers, above mentioned n, m, k, l. The second line, consisting of n integers, is the waveform S recorded by the receiver. The k following lines, each of which consists of m integers, are the waveforms of the k special words $W_1 \dots W_k$.

Output

For each test case, print the minimum *Similarity Value* in a line. You are not required to print the corresponding *Decoding Result* because the problem writer is nice.

Constraints:

$T \le 15$,
$0 < n \le 10^5$,
$0 < m \le 3 \cdot 10^4$,
$0 < k \le 5$,
$l \cdot m \le n$,
All values of the waveforms are within range [0,100].

Sample Input

```
3
10 1 2 3
0 1 2 3 4 5 6 7 8 9
7
8
10 3 2 2
0 1 2 3 4 5 6 7 8 9
3 4 5
1 2 3
10 3 2 3
0 1 2 3 4 5 6 7 8 9
6 7 8
1 2 3
```

Sample Output

```
1
3
15
```