Variance Challenge

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

Kevin has recently learned the definition of variance. For an array a of length n, the variance of a is defined as follows:

- Let $x = \frac{1}{n}\sum_{i=1}^{n} a_i$, i.e., x is the mean of the array a;
- Then, the variance of a is
$$V(a) = \frac{1}{n}\sum_{i=1}^{n}(a_i - x)^2.$$

Now, Kevin gives you an array a consisting of n integers, as well as an integer k. You can perform the following operation on a:

- Select an interval $[l, r]$ $(1 \le l \le r \le n)$, then for each $l \le i \le r$, increase a_i by k.

For each $1 \le p \le m$, you have to find the minimum possible variance of a after exactly p operations are performed, independently for each p.

For simplicity, you only need to output the answers multiplied by n^2. It can be proven that the results are always integers.

Input

Each test contains multiple test cases. The first line of the input contains a single integer t $(1 \le t \le 100)$ — the number of test cases. The description of test cases follows.

The first line of each test case contains three integers n, m, and k $(1 \le n, m \le 5000$, $n \cdot m \le 2 \cdot 10^4$, $1 \le k \le 10^5)$ — the length of the array a, the maximum number of operations, and the number you add to a_i each time, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^5)$ — the elements of the array a.

It is guaranteed that the sum of $n \cdot m$ over all tests does not exceed $2 \cdot 10^4$.

Output

For each test case, output m integers in a single line, the p-th integer denoting the minimum possible variance of a when exactly p operations are performed, multiplied by n^2.

Example

standard input	standard output
9 3 2 1 1 2 2 3 2 2 1 2 2 10 2 1 10 1 1 1 1 10 1 1 1 1 6 8 2 1 1 4 5 1 3 8 8 7 20 43 24 2 4 3 20 43 8 8 3 20 43 24 2 4 3 20 43 10 12 1 5 3 3 5 4 1 8 1 1 1 13 10 100000 1 2 3 4 5 6 7 8 9 10 11 5 4 10 5 10000 2308 9982 4435 3310 100000 9 7 8100 1919 100000	0 0 2 2 1161 1024 53 21 21 5 5 5 5 5 10608 6912 4448 3104 1991 1312 535 304 13248 11184 9375 7815 6447 5319 4383 3687 385 316 269 224 181 156 124 101 80 56 41 29 1486 1486 1486 1486 1486 1486 1486 1486 1486 1486 134618047140 119919447140 107020847140 93922247140 82623

Note

In the first test case:

- For $p = 1$, you can perform the operation on $[1, 1]$, changing a from $[1, 2, 2]$ to $[2, 2, 2]$. Since all of the elements are equal, the variance is equal to 0.
- For $p = 2$, you can perform the operation on $[1, 3]$ and then $[1, 1]$, changing a from $[1, 2, 2]$ to $[2, 3, 3]$ to $[3, 3, 3]$. Since all of the elements are equal, the variance is equal to 0.

In the second test case, some possible optimal choices are:

- $p = 1$: $[\underline{1}, 2, 2] \to [3, 2, 2]$;
- $p = 2$: $[1, \underline{2, 2}] \to [\underline{1}, 4, 4] \to [3, 4, 4]$.

In the third test case, some possible optimal choices are:

- $p = 1$: $[10, \underline{1, 1, 1, 1, 10, 1, 1, 1, 1}] \to [10, 2, 2, 2, 2, 11, 2, 2, 2, 2]$;
- $p = 2$: $[10, 1, 1, 1, 1, 10, \underline{1, 1, 1, 1}] \to [10, \underline{1, 1, 1, 1}, 10, 2, 2, 2, 2] \to [10, 2, 2, 2, 2, 10, 2, 2, 2, 2]$.

In the eighth test case, the optimal choice for all p is to perform the operation on the whole array p times.