

A cellular network is a radio network made up of a number of cells each served by a base station located in the cell. The base station receives call signals from mobile users (*mobiles*) in the cell it serves, which then connects the calls to the wired land-line telephone network. When a call is requested to connect to a mobile, the cellular network must know in which cell the mobile is located so that the call is routed to the base station of the cell appropriately.

Mobiles move from once cell to another in a cellular network. Whenever a mobile reports its new cell as it crosses boundaries of cells, the cellular network would know its exact cell at any time and finding (*paging*) the mobile becomes a trivial task. But it is usually infeasible for a mobile to report its new location each time it enters a new cell because of the insufficiencies of resources such as the radio bandwidth. But normally at the time of a call arrival, the cellular network knows a limited number of cells where the mobile is located. In this situation, a lot of paging strategies are developed to locate a mobile efficiently. The ultimate goal of paging strategies is to minimize both the time delay and cost of paging until the mobile is found.

Now we define our problem formally. The *location area* is the set of n cells $C = \{c_1, c_2, \dots, c_n\}$ such that the mobile is guaranteed to be in one of these cells at the time of a call arrival. Suppose that it is possible to page any subset of these n cells in a unit of time (*paging rounds*) and find out if the mobile is located in one of the cells paged. The fastest strategy to find the cell where the mobile is located is to page all the n cells in the first and only round. However this strategy uses a lot of wireless bandwidth.

In many cases, the cellular network knows about the whereabouts of the mobile. This knowledge can be modeled with n probability values, where the probability of the mobile being present in a cell can be estimated for each of these n cells at the time of a call arrival. Let p_i be the probability that the mobile is located at the cell c_i and all the probabilities are independent. A sequential paging strategy is to page the cells sequentially in n paging rounds terminating once the mobile is found. Then the average cost of paging (number of cells paged), \bar{C} , and the average paging delay (number of paging rounds) in locating the mobile, \bar{D} , can be expressed as follows:

$$\bar{C} = \sum_{i=1}^n (i \times p_i), \quad \bar{D} = \sum_{i=1}^n (i \times p_i).$$

The *parallel paging strategy* is to page the cells in a collection of cells simultaneously. Sequential paging strategy has lower paging cost than parallel paging strategy, but at the expense of larger paging delay. The method of parallel paging is to partition the cells in a location area into a series of indexed groups referred to as *paging zones*. Let Z_1, Z_2, \dots, Z_w be the partition of the location area C (i.e., a partition of C into w groups), where each Z_i is non-empty and corresponds to a distinct paging zone. When a call arrives for a mobile, the cells in the first paging zone Z_1 are paged simultaneously in the first round and then if the mobile is not found in the first round of paging, all the cells in the second paging zone Z_2 are paged, and so on. Let the number of cells in the paging zone Z_i be denoted by $n_i = |Z_i|$, and let π_i be the corresponding zone probabilities of the users in the paging zone Z_i , where $\pi_i = \sum_{c_j \in Z_i} p_j$. Then the average cost of paging (number of cells paged), \bar{C} , and the average paging delay (number of paging rounds) in locating the mobile, \bar{D} , can be expressed as follows:

$$\bar{C} = \sum_{i=1}^w \left(\sum_{j=1}^i n_j \right) \pi_i, \quad \bar{D} = \sum_{i=1}^w (i \times \pi_i).$$

In parallel paging strategy, there is a tradeoff between bandwidth for time. For example, we increase the number of paging zones, then the paging cost could be decreased. If we decrease the number of paging zones, then the paging cost could be increased. Furthermore, for a fixed number w of paging zones, the paging cost could be different to the strategies how the cells in location area are partitioned.

For example, there are $n = 5$ cells in a location area $C = \{c_1, c_2, \dots, c_5\}$ and the probability of each cell in C are as follows:

c_i	c_1	c_2	c_3	c_4	c_5
p_i	0.3	0.05	0.1	0.3	0.25

If the cells in C are partitioned into two paging zones $Z_1 = \{c_1, c_2, c_3\}$, $Z_2 = \{c_4, c_5\}$, the average cost of paging, \bar{C} , and the average paging delay in locating the mobile, \bar{D} , are:

$$\bar{C} = n_1\pi_1 + (n_1 + n_2)\pi_2 = 3(0.3 + 0.05 + 0.1) + (3 + 2)(0.3 + 0.25) = 3 \times 0.45 + 5 \times 0.55 = 4.1$$

$$\bar{D} = 1\pi_1 + 2\pi_2 = 1(0.3 + 0.05 + 0.1) + 2(0.3 + 0.25) = 1 \times 0.45 + 2 \times 0.55 = 1.55$$

If the cells in C are partitioned into two paging zones $Z_1 = \{c_1, c_4\}$, $Z_2 = \{c_2, c_3, c_5\}$, the average cost of paging, \bar{C} , and the average paging delay in locating the mobile, \bar{D} , are:

$$\bar{C} = n_1\pi_1 + (n_1 + n_2)\pi_2 = 2(0.3 + 0.3) + (3 + 2)(0.05 + 0.1 + 0.25) = 2 \times 0.6 + 5 \times 0.4 = 3.2$$

$$\bar{D} = 1\pi_1 + 2\pi_2 = 1(0.3 + 0.3) + 2(0.05 + 0.1 + 0.25) = 1 \times 0.6 + 2 \times 0.4 = 1.4$$

Given the number of cells in a location area C , the probabilities of each cell that a mobile is located at the cell, and the fixed number w of paging zones, write a program to partition the cells in C into w paging zones such that the average cost of paging to find the location of the mobile is minimized.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two lines. The first line of each test case contains two integers. The first integer, n , is the number of cells in a location area, and the second integer, w , is the number of paging zones, where $1 \leq w \leq n \leq 100$. The second line of each test case contains n integers u_1, u_2, \dots, u_n , where the probability p_i for each cell c_i in C is $p_i = u_i / (u_1 + u_2 + \dots + u_n)$. All integers in the second line are between 1 and 10,000.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain the minimum average cost of paging to find the location of the mobile. The output should have a precision of exactly 4 digits after decimal point. You may round to the 4 digits after decimal point or round off at the 4-th digit after decimal point.

Sample Input

```
2
5 2
30 5 10 30 25
5 5
30 5 10 30 25
```

Sample Output

```
3.2000
2.3000
```