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 one cell to another in a cellular network. Whenever a mobile reports its new location to the cellular network, the network must know 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 is given in the first line of the input. Each test case consists of two lines. The first line of a test case is the number of cells "n" that a mobile can be 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. 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 cost and cost of paging until the mobile is found. Then the average cost of paging (number of cells paged), \( C_{avg} \), and the average paging delay (number of paging rounds) in locating the mobile, \( D \), can be expressed as follows:

\[
C_{avg} = \sum_{i=1}^{n} \sum_{j=1}^{w} c_{ij} \quad D = \sum_{i=1}^{n} \sum_{j=1}^{w} \left( \frac{1}{p_{i}} \right)
\]

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 \( Z_{1}, Z_{2}, \ldots, Z_{w} \) be the partition of the location area \( C \) (i.e., a partition of \( C \) into \( w \) groups), where each \( Z_{j} \) 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 \( Z_{j} \) be denoted by \( n_{j} = |Z_{j}| \) and let \( p_{i} \) be the corresponding zone probabilities of the users in the paging zone \( Z_{j} \), where \( p_{i} = \sum_{j=1}^{w} p_{ij} \). Then the average cost of paging (number of cells paged), \( C \), and the average paging delay (number of paging rounds) in locating the mobile, \( D \), can be expressed as follows:

\[
C = \sum_{i=1}^{n} \sum_{j=1}^{w} c_{ij} \\
D = \sum_{i=1}^{n} \sum_{j=1}^{w} \left( \frac{1}{p_{i}} \right)
\]

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, then 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 \( C = \{c_{1}, c_{2}, \ldots, c_{5}\} \) and the probability of each cells in \( C \) are as follows:

\[
\begin{align*}
&c_{1} &c_{2} &c_{3} &c_{4} &c_{5} \\
&0.3 &0.05 &0.1 &0.3 &0.25 
\end{align*}
\]

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, \( C \), and the average paging delay in locating the mobile, \( D \), are:

\[
C = n_{1}c_{1} + n_{2}c_{2} = (0.3 + 0.05 + 0.1) + (0.3 + 0.05 + 0.1) = 0.85 + 0.85 = 1.7 \\
D = 1n_{1} + 2n_{2} = 1 \times 0.85 + 2 \times 0.85 = 3.7
\]

If the cells in \( C \) are partitioned into three paging zones \( Z_{1} = \{c_{1}, c_{2}\}, Z_{2} = \{c_{3}\}, Z_{3} = \{c_{4}, c_{5}\} \), the average cost of paging, \( C \), and the average paging delay in locating the mobile, \( D \), are:

\[
C = n_{1}c_{1} + n_{2}c_{2} + n_{3}c_{3} = (0.3 + 0.05 + 0.1) + (0.3 + 0.05 + 0.1) + 0.3 = 1.1 + 1.1 + 0.3 = 2.5 \\
D = 1n_{1} + 2n_{2} + 3n_{3} = 1 \times 0.85 + 2 \times 0.85 + 3 \times 0.3 = 0.85 + 1.7 + 0.9 = 3.45
\]

Given the number of cells in a location area \( C \), the probabilities of each cells 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 \( n \) positive integers \( a_{1}, a_{2}, \ldots, a_{n} \), where the probability \( p_{i} \) for each cell \( c_{i} \) in \( C \) is \( p_{i} = a_{i}/\sum a_{i} \). All integers in the second line are between 1 and 10000.

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
10 30 25
5 5
30 5 10 30 25
Sample Output
3.2000
2.3000