Sharif University of Technology (SUT) has provided a special bus service to take back the interested employees to their homes each day. To use this service, employees should register online. At the beginning of each day, the list of people using the service on that day is posted online making sure that the capacity of the bus is met.

The rent of the bus on each day is p Toomans independent of the number of people in the bus, and as a rule accepted by all, only one of the people in that bus should pay the bus rent to the bus driver. The name of the person responsible to pay the rent is also announced daily.

Your job is to write a program to help SUT select the person responsible for the bus payment in each day such that the assignments become "fair" as defined next.

Assume L_1, L_2, ..., L_d are the lists of employees in the bus for day 1 to day d and assume that n_i is the size of L_i. If a person named A uses the bus for k days t_1, t_2, ..., and t_k, his correct share of the bus rents is $P_A = p(1/n_{t_1} + 1/n_{t_2} + \ldots + 1/n_{t_k})$ Toomans. If A is selected to pay the bus rent for r times, he will pay $Q_A = r \times p$ Toomans which is $E_A = Q_A - P_A$ Toomans more than his share. The "unfairness" of an assignment is defined to be the maximum of all E_A. A "fair" assignment is the one with the minimum unfairness.

Input

There are multiple test cases in the input. For each test case, the first line contains three positive integers n, d and p ($n, d \leq 500, p \leq 10^9$) where n, d and p are the number of employees, the number of days and the rent of the bus for each day. The information of each day comes in the next d lines; one line per day. Each line starts with the number of employees using the bus on that day followed by a list of employee IDs which are integer numbers in the range $[1, n]$. For ease of computation, the bus rent p is chosen such that the share of employees in each day is an integer number. The input terminates with a line containing '`0 0 0`' which should not be processed.

Output

For each test case, output a line containing the unfairness of a fair assignment.

Sample Input

```
3 2 1000
2 1 2
2 1 3
4 4 3000
2 1 2
2 1 3
2 2 3
3 2 3 4
0 0 0
```

Sample Output

```
500
2000
```