Booklets

Bob has a difficult job. He must distribute advertising booklets for extra school activities in different schools. The booklets have different number of pages. Bob has a list with the number of pages of each booklet and the number of schools that he must visit. He has to distribute the booklets such that each school gets a number of booklets equal to either the lower integer part (LIP),or the upper integer part (UIP) of the number of booklets divided by the number of schools. Poor Bob must obey other rules too. He must distribute all the **UIP** number of booklets first and then the **LIP** number of booklets.

Any booklet **A** that is distributed to a school **S_i** must have fewer or at most an equal number of pages that any other booklet B that is distributed to a school **S_j**, if **S_i** gets the booklets before **S_j** (i.e if **i** < **j** then **pages(A)** <= **pages(B)**). When Bob distributes the booklets to a school he must distribute them in the same relative order in which they are on his list.

Moreover, he must distribute them very fast. When he comes back to the advertising company his boss verifies if he accomplished well his task, by asking him the number of pages of the first booklet distributed to a specific school, following the order in which Bob visited the schools (starting with 0). Difficult job, isn't it? Can you help him?

Input

The input starts with a line containing a single integer **t** <= 20, the number of test cases. **t** test cases follow.

Each data set in the input stands for a particular set of bookets. For each set of booklets the input contains the number of schools, the school specified by Bob's boss, the number of booklets (**less than 3000**), the number of pages of each booklet (fits in integer). White spaces can occur freely between the numbers in the input. The input data are correct.

Output

For each set of data the program prints the result to the standard output on a separate line. The solution is represented by the number of pages of the first booklet distributed to the specified school.

Example

Input:

```
1
3
2
7
3 5 9 1 11 14 2
```

Output:

```
11
```