10412   Big Big Trees

There are n trees arranged in a straight line in the forest, the adjacent two trees are m meters apart from each other. The i-th tree is h_i meters high, and at every integer height x $(1 \le x \le h_i)$, the tree has two big leaves with a length of $l_{i,j}$ on both sides. The two leaves are left-right symmetrical, so each tree is left-right symmetrical too. The picture below shows two trees. Note that the longest leaf should be shorter than $m/2$ meters. So every two leaves from different trees can NEVER overlap in left-to-right direction.
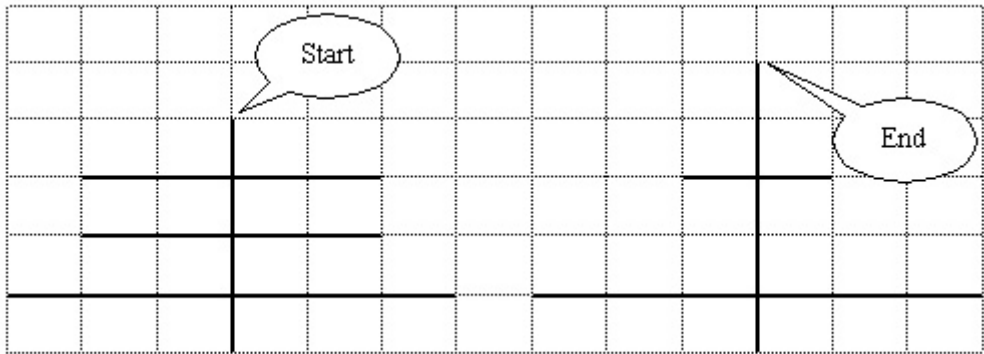
A monkey from the top of the 1st tree wants to reach to top of the last tree. He may climb up and down, and he may walk from left to right on the leaves or on the ground. He may also jump from the RIGHT ENDPOINT of a leaf to the LEFT ENDPOINT of another leaf on the next tree (Warning: the monkey cannot jump from or land inside a leaf or the ground! What's more, he jumps along a straight line, so the line should not contain any points of any other leaves). But he may do that only if the distance between the two endpoints are not longer than k meters. Clearly, the monkey should always use exactly $n-1$ jumps, but he may wall on different route in order to minimize the total distance he walks (NOT climbs). Help him to calculate the minimal total distance walked.

Input

The first line of the input is a single integer t $(1 \le t \le 10)$, indicating the number of test cases. Each case begins with a line containing three integers n, m, k $(1 \le n, m, k \le 1000)$, indicating the number of trees, the distance between two adjacent trees, and the maximal allowed jump distance. There are n lines following. Each line describes a tree. The first integer h $(1 \le h \le 20)$ is the height of the tree. There are h integers following: $l_1, l_2, \ldots, l_h$ $(0 \le l_i < m/2, i = 1, 2, \ldots, n)$, indicating the lengths of the leaves.

Output

For each case, print a number d in a single line indicating the minimal total distance the monkey walks.

Sample Input

```
2
2 7 3
4 3 2 2 0
```

```
5 3 0 1 0 0
3 50 40
4 15 3 16 10
8 12 12 12 21 12 15 6 14
13 15 23 20 18 14 1 21 9 9 18 23 10 4
```

Sample Output

```
5
28
```