

Professor Heickal absolutely hates cycles. So when it comes to graphs, he is partial to trees and forests. Not surprisingly, his favorite band is ‘Porcupine Tree’, favorite movie is the ‘Forrest Gump’ and he loves eating Black Forests. Anyway, in one evening, I went to him to discuss about problems concerning people, country, the world and the humankind.

In the middle of our discussion, he suddenly said, “See! Everybody loves cycles! People like to live the life their ancestors lived, the world leaders never learn from their mistakes, people cut down trees and destroy forests all the time and Professor Hawlader keeps on giving people amulets to change their fortunes! These kinds of things are recurring all the time! You got to stop these recurrences by putting a base case. Scientists say, Dynamic Programming is very important for life. Everyone knows that trees, forests and directed acyclic graphs are suitable for DP...”. At that point, he noticed that I was nervously checking my watch. He changed the topic and said, “Let’s solve an interesting problem! And it’s about... trees!”

The problem was like this: Given the number of nodes at different depths of a rooted tree, count the number of valid trees possible with that configuration. Formally, you will be given an array  $C$  with  $C_i = \text{number of nodes at depth } i$  (explanations of the terms is given afterwards). Each node of the tree is labeled with a unique integer. The nodes are labeled in this way:

$$\text{Let, } S_i = C_0 + C_1 + \dots + C_i$$

The root is labeled with 1.

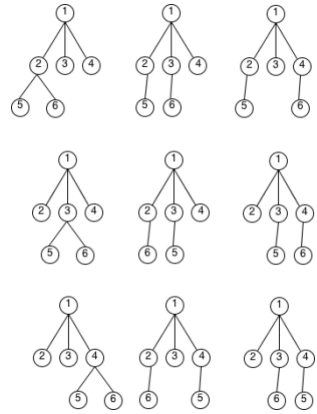
The nodes at depth 1 are labeled by integers from 2 to  $S_1$  (inclusive) . . . . .

The nodes at depth  $d$  are labeled by integers from  $S_{d-1} + 1$  to  $S_d$  (inclusive)

You’ll need to find out the number of  $k$ -ary trees with  $C$  array.

A tree is a connected graph without a cycle. A tree is called a rooted tree if one vertex has been designated the root, in which case the edges have a natural orientation, away from the root. The depth of a node is the length of the path to its root. A  $k$ -ary tree is a rooted tree in which each node has no more than  $k$  children.

Anyway, as I was severely disoriented by that discussion with the professor, I could not solve the problem that time. What about you?



## Input

The first line of the input will contain the number  $T$ , the number of test cases. It will be followed by  $T$  sets of inputs. Each set of test case will have two lines. First line will contain pair of integers  $d$ , depth of the tree and  $k$ , as defined in the statement. In the next line, there will be  $d + 1$  integers, giving the  $C$  array where  $i$ -th integer will be the value of  $C_i$ .

### Constraints:

- $1 \leq T \leq 256$
- $0 \leq d \leq 512$
- $0 \leq k \leq 8$
- $0 \leq C_i \leq 512$  ( $1 \leq i \leq d$ ) and  $C_0 = 1$

## Output

For each set of input, print the output in the format ‘Case  $X$ :  $Y$ ’ (here,  $X$  is the serial of the input and  $Y$  is the answer) in a line. As the output can be very large, print the answer modulo 1000000009.

## Sample Input

```
2
2 3
1 3 2
8 8
1 4 10 14 17 24 26 26 30
```

## Sample Output

```
Case 1: 9
Case 2: 23879694
```