The Kth BST

Time Limit: 1 Second      Memory Limit: 32768 KB

**Definition:** A *binary tree* is a finite set of nodes that is either empty or consists of a root and two disjoint binary trees called the left subtree and the right subtree.

**Definition:** A *binary search tree*(BST) is a binary tree. It may be empty. If it is not empty, it satisfies the following properties:

1. Every elements has a key, and no two elements have the same key, that is, the keys are unique.
2. The keys in a nonempty left subtree must be smaller than the key in the root of the subtree.
3. The keys in a nonempty right subtree must be larger than the key in the root of the subtree.
4. The left and right subtrees are also binary search trees.

In this problem, we just care about the *Preorder Traversal* of a BST. Here is the pseudocode for *Preorder Traversal*:

void preorder(tree\_pointer ptr)

/\* preorder tree traversal \*/

{

if (ptr) {

printf("%d", ptr->data);

preorder(ptr->left\_child);

preorder(ptr->right\_child);

}

}

Now, you are given **n**, the number of nodes in a BST, and the nodes of the BST are consist of the first **n** lowcase letters. Of course, more than one BST can be constructed except when **n** is 1. You task is to sort there BST's according to their *preorder* representations, and gives out the **K**th BST.

For example, when **n** is 2, there are two BST's can be constructed, as following:

a b

\ /

b a

Their *preorder* representations are: **ab** and **ba**, so the first one is **ab**, and the second one is **ba**.

**Input**

There are multiple test cases in this problem. The input is terminated by **EOF**.

For each test case, there are two inputs: **n** and **K**, representing the number of nodes in the BST, and the index of the BST you need to output.

*Note:*

* **n** is between 1 and 19
* **K** is between 1 and the number of ways to construct the BST

**Output**

For each input, you should first output the **K**th *preorder* representation of the BST. Next, for each node (in the order a, b, c, ...), output it first, then output the left sub node (output **\*** if not exist) and the right sub node (output **\*** if not exist), seperated by a single blank space. **K** will not be greater than the number of representations of BST given **n** nodes. Output a blank line between two test cases.

**Sample Input**

2 2

4 9

**Sample Output**

ba

a \* \*

b a \*

cbad

a \* \*

b a \*

c b d

d \* \*

Author: **DAI, Wenbin**  
Source: **Zhejiang Provincial Programming Contest 2006, Preliminary**