Christmas Tree

Chirag is a pure Desi boy. And his one and only dream is to meet Santa Claus. He decided to decorate a Christmas tree for Santa on coming Christmas. Chirag made an interesting Christmas tree that grows day by day.

The Christmas tree is comprised of the following
Parts
Stand

Each Part is further comprised of Branches. Branches are comprised of Leaves.

How the tree appears as a function of days should be understood. Basis that print the tree as it appears on the given day. Below are the rules that govern how the tree appears on a given day.

Write a program to generate such a Christmas tree whose input is number of days.

Rules:

If tree is one day old you cannot grow. Print a message “You cannot generate christmas tree”
Tree will die after 20 days; it should give a message “Tree is no more”
Tree will have one part less than the number of days. E.g.
On 2nd day tree will have 1 part and one stand.
On 3rd day tree will have 2 parts and one stand
On 4th day tree will have 3 parts and one stand and so on.
Top-most part will be the widest and bottom-most part will be the narrowest.
Difference in number of branches between top-most and second from top will be 2
Difference in number of branches between second from top and bottom-most part will be 1

Input Format:

First line of input contains k - the number of inputs
The next k lines denote the number of days N

Output Format:
Print Christmas Tree for given N

OR

Print “You cannot generate christmas tree” if N <= 1

OR

Print “Tree is no more” if N > 20

Constraints:
0<= N <=20

Example:

Input:
k = 2
N = 1
N = 2

Output:

You cannot generate christmas tree
```
  *
 ***
*****
  *
  *
```

Code :

```
import java.util.*;

class Main{
```

```
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();

        for(int i = 1 ; i <= n ; i++){
            sc.nextLine();
            int k = sc.nextInt();

            if(k <= 1)  System.out.println("You cannot generate christmas
tree");
            else if( k > 20)  System.out.println("Tree is no more");
            else{
            printFullTriangel(k+1);
            printPartialTriangel(k);
            printStand(k);
            }
        }
    }

    static void printFullTriangel(int k){
        for(int i = 1 ; i <= k ; i++){
            for(int n=1 ; n <= (k-i) ; n++)  System.out.print(" ");
            for(int m=1 ; m <= ((2*i)- 1) ; m++) System.out.print("*");
            System.out.println();
          }
    }
    static void printPartialTriangel(int k){
        int offset = 1;
        for(int i = k-1 ; i >=2 ; i--){
            partial(i , offset);
            offset++;
          }
    }
    static void partial(int k, int offset){
        for(int i = 1 ; i <= k ; i++){
            for(int m=1 ; m <= offset ; m++)  System.out.print(" ");
            for(int n=1 ; n <= (k-i) ; n++)  System.out.print(" ");
            for(int p=1 ; p <= ((2*i)+ 1) ; p++) System.out.print("*");
```

```
            System.out.println();
          }
    }
    static void printStand(int k){
        for(int i = 1 ; i <= 2 ; i++){
            for(int j = 1 ; j <= k ; j++) System.out.print(" ");
            System.out.print("*");
            System.out.println();
        }
    }

}
```