```
/**Day 83 coding Statement :

Given a complete binary tree with the height of H, we index the nodes respectively top-down and left-right from 1. The i-th node stores a positive integer Vi. Define Pi as follows: Pi=Vi if the i-th node is a leaf, otherwise Pi=max(Vi*PL, Vi*PR), where L and R are the indices of the left and right children of i, respectively. Your task is to caculate the value of P1.

**/
import java.math.BigInteger;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
class Main {

public static BigInteger MOD = new BigInteger ("1000000007");
public static BigInteger pr (int i, int len, BigInteger v[]) {
if (2 * i > len)
return v[i];
return pr(2 * i, len, v).max(pr(2 * i + 1, len, v)).multiply(v[i]);
}
public static void main (String [] ar) throws IOException {
BufferedReader br = new BufferedReader (new InputStreamReader(System.in));
int n, len;
BigInteger v[];
String tmp[];
while ((n = Integer.parseInt(br.readLine())) != 0) {
len = (1 << n) - 1;
v = new BigInteger[len + 5];
tmp = br.readLine().split(" ");
for (int i = 1; i <= len; i++)
v[i] = new BigInteger(tmp[i - 1]);
System.out.println(pr(1, len, v).mod(MOD));
}
}
}
```

Talent Ba