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 V_i. Define P_i as follows: P_i=V_i if the i-th node is a leaf, otherwise P_i=max(V_i*P_L, V_i*P_R), where L and R are the indices of the left and right children of i, respectively. Your task is to caculate the value of P_1.

Input

There are several test cases (fifteen at most), each formed as follows:

- The first line contains a positive integer H (H ≤ 15).
- The second line contains 2^H-1 positive integers (each having a value of 10^9 at most), the i-th integer shows the value of V_i.

The input is ended with H = 0.

Output

For each test case, output on a line an integer which is the respective value of P_1 found, by modulo of 1,000,000,007.

Sample Input

2

1 2 3

3

3 1 5 2 6 4 7

0

Sample Output

3

105

Explanation

```
    3

   / \

  /   \

  1    5

/ \  / \

2  6 4  7
```

C

```
#include<stdio.h>
#define mod 1000000007
long long a[32768];
double b[32768];
inline long long readint()
{
  int r=0;
  char p=getchar_unlocked();
  for(;p<33;){p=getchar_unlocked();};
  while(p>32)
  {
  r = (r << 3) + (r << 1) + (p - '0');
  p=getchar_unlocked();
  }
  return r;
};
int main()
{
  int h,i,n;
  while(1)
  {
    n=readint();
    //scanf("%d",&h);
    if(n==0)break;
    h=(1<<n)-1;
    for(i=1;i<=h;i++)
    {
      b[i]=a[i]=readint();
    }
    for(i=h/2;i>=1;i--)
    {
      if(b[2*i]>b[2*i+1])
      {
        a[i]*=a[2*i];
        b[i]*=b[2*i];
      }
      else
      {
        a[i]*=a[2*i+1];
        b[i]*=b[2*i+1];
      }
      a[i]%=mod;
    }
    printf("%lld\n",a[1]);
  }
  return 0;
}
```

C++

```
#include<bits/stdc++.h>
using namespace std;
#define q 1000000007
int main()
{
  long long int t,n,i,j;
  cin>>t;
  while(t!=0)
  {
    n=(1<<t) - 1;


    long long int a[n+1];
    double b[n+1];
    for(i=1;i<=n;i++){cin>>a[i];b[i]=a[i];}


    for(i=n/2;i>0;i--)
    {
        if(b[2*i +1]>b[2*i]){a[i]=a[i]*a[2*i+1];b[i]=b[i]*b[2*i+1];}
        else {a[i]=(a[i]*a[2*i]);b[i]=(b[i]*b[2*i]);}


        a[i]%=q;


    }
    cout<<a[1]<<endl;
    cin>>t;
  }
  return 0;
}
```

Java

```
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));
        }
    }
}
```

Python

```
z=0
while z==0:
  h=int(input())
  if h==0:
      break
  n=(1<<h)-1
  p=[0]*(n+10)
  v=[ int(x) for x in input().split() ]
  for i in range(n-1,-1,-1):
      if 2*i+1 >=n:
          p[i]=v[i]
      else:
          p[i]=v[i]*max(p[2*i+1],p[2*i+2])
  ans=p[0]%1000000007
  print(ans)
```