一、编程题

ACM：信道分配

题目描述：算法工程师小明面对着这样一个问题，需要将通信用的信道分配给尽量多的用户：

信道的条件及分配规则如下：

1) 所有信道都有属性："阶"。阶为r的信道的容量为2^r比特；
2) 所有用户需要传输的数据量都一样：D比特；
3) 一个用户可以分配多个信道，但每个信道只能分配给一个用户；
4) 只有当分配给一个用户的所有信道的容量和>=D，用户才能传输数据；

给出一组信道资源，最多可以为多少用户传输数据？

输入描述：第一行，一个数字R。R为最大阶数。
0<=R<20

第二行，R+1个数字，用空格隔开。
代表每种信道的数量Ni。按照阶的值从小到大排列。
0<=i<=R, 0<=Ni<1000.

第三行，一个数字D。
D为单个用户需要传输的数据量。
0<D<1000000

输出描述：一个数字，代表最多可以供多少用户传输数据。

补充说明：

示例1
输入：5
10 5 0 1 3 2
30
输出：4
说明：最大阶数为5.
信道阶数：0 1 2 3 4 5
信道容量：1 2 4 8 16 32
信道个数：10 5 0 1 3 2
单个用户需要传输的数据量为30

可能存在很多分配方式，举例说明：
分配方式1：
1) 32*1 = 32
2) 32*1 = 32
3) 16*2 = 32
4) 16*1 + 8*1 + 2*3 = 30
剩下2*2 + 1*10=14不足以再分一个用户了。

分配方式2：
1) 16*1 + 8*1 + 2*3 = 30
2) 16*1 + 2*2 + 1*10 = 30
3) 32*1 = 32
4) 32*1 = 32
剩下16*1=16不足以再分一个用户了。

分配方式3：
1) 16*1 + 8*1 + 2*3 = 30
2) 16*1 + 2*2 + 1*10 = 30
3) 16*1 + 32*1 = 48
4) 32*1 = 32
恰好用完。

虽然每种分配方式剩下的容量不同，但服务的用户数量是一致的。因为这个问题中我们只关心服务的用户数，所以我们认为这些分配方式等效。

代码：

```
import java.util.Arrays;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int r = Integer.parseInt(sc.nextLine());
        Integer[] arr = Arrays.stream(sc.nextLine().split("\\s+")).map(Integer::parseInt).toArray(Integer[]::new);
        int d = Integer.parseInt(sc.nextLine());

        int count = 0;
        int index = 0;
        HashMap<Integer, Integer> dmap = new HashMap<>();
        while (index < arr.length) {
            dmap.put((int) (Math.pow(2, index)), arr[index]);
            index += 1;
        }
        for (Integer k : dmap.keySet()) {
            if (k >= d) {
                count += dmap.getOrDefault(k, 0);
                dmap.put(k, 0);
            }
        }
        int cap = 0;
        for (int k : dmap.keySet()) {
            cap += k * dmap.getOrDefault(k, 0);
        }
        count += (cap / d);

        System.out.println(count);
    }
}
```