

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 $\geq D$ ，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。

$0 \leq R < 20$

第二行， $R+1$ 个数字，用空格隔开。

代表每种信道的数量 N_i 。按照阶的值从小到大排列。

$0 \leq i \leq R, 0 \leq N_i < 1000$ 。

第三行，一个数字 D 。

D 为单个用户需要传输的数据量。

$0 < D < 10000000$

输出描述：

一个数字，代表最多可以供多少用户传输数据。

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。

```
def max_num_user(R, Ni, D):
    idx = 0
    max_user = 0
    mapping = {}

    n = R + 1

    while (idx < n):
        mapping[2 ** idx] = Ni[idx]
        idx += 1
    for item in mapping.keys():
        if item >= D:
            max_user += mapping[item]
            mapping[item] = 0

    x = 0
    for j in mapping.keys():
        x += j * mapping[j]
    max_user += int(x // D)

    return max_user

# 输入
R = int(input())
Ni = list(map(int, input().split()))
```

```
D = int(input())
```

```
print(max_num_user(R, Ni, D))
```