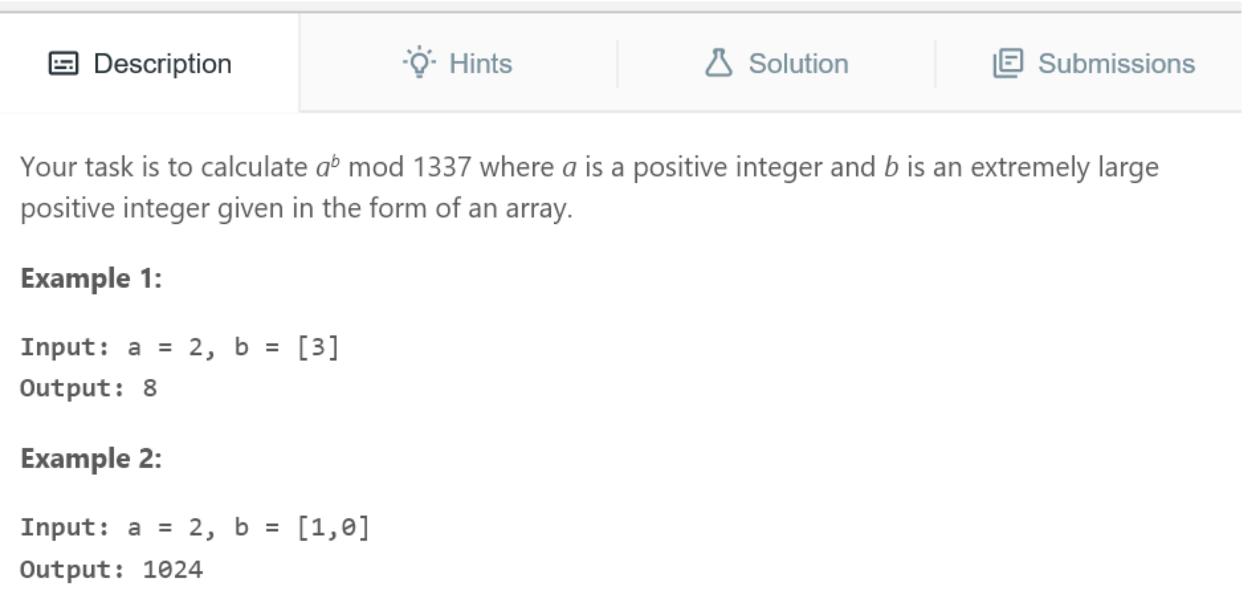


Your task is to calculate a^b mod 1337 where a is a positive integer and b is an extremely large positive integer given in the form of an array.

Example 1:

```
Input: a = 2, b = [3]
Output: 8
```

Example 2:

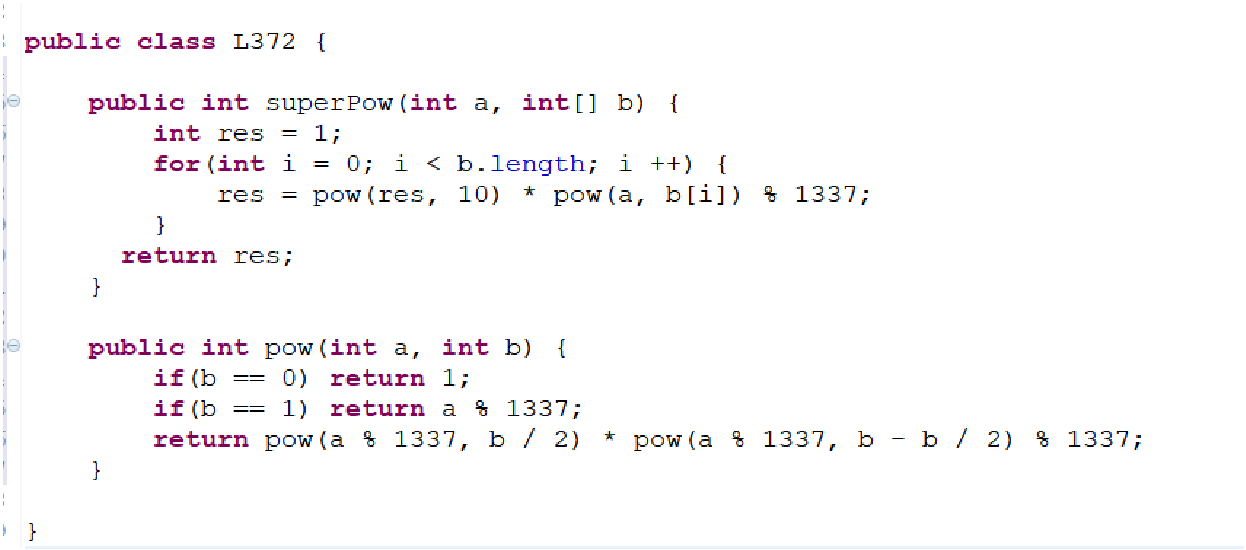
```
Input: a = 2, b = [1,0]
Output: 1024
```

思路

由于直接ab%1337会溢出，b本身也会溢出，它是一个超大数，所以ab%1337我们变换成 (a%1337)*(a%1337) %1337*(a%1337)%1337......这样处理。具体看代码吧。一目了然。

算法需要利用的恒等式 (a*b)%c = (a%c)*(b%c)%c，**证明如下**:

```
 设a/c=m,则mc+a%c =a;设b/c=n,则nc+b%c =b;于是
  (a*b)%c
= { (mc+a%c)*(nc+b%c ) }%c
= {mcnc+(nc)*(a%c)+(mc)*(b%c)+(a%c)*(b%c)} % c (其中 mcnc+(nc)*(a%c)+(mc)*(b%c)可以整除c)
= (a%c)*(b%c)%c
```



```java
public class L372 {

    public int superPow(int a, int[] b) {
        int res = 1;
        for(int i = 0; i < b.length; i ++) {
            res = pow(res, 10) * pow(a, b[i]) % 1337;
        }
      return res;
    }

    public int pow(int a, int b) {
        if(b == 0) return 1;
        if(b == 1) return a % 1337;
        return pow(a % 1337, b / 2) * pow(a % 1337, b - b / 2) % 1337;
    }

}
```