Consider,

$$\text{MODEXP} = \left\{ \begin{array}{l} \langle a,b,c,p\rangle \mid a,b,c \text{ and } p \text{ are binary integers} \\ \text{such that } a^b \equiv c \pmod p \end{array} \right\}$$

A polynomial time algorithm *M* for MODEXP is as follows:

$M =$ "On input $< a,\ b,\ c, p >$, where $a,\ b,\ c$ and p are binary integers.

- Calculate $x\ =\ a \bmod p$, initialize y to 1 and i to 0.
- For $b = b_n b_{n-1} ... b_1 b_0$, do the following $n+1$ times:
- if $b_i = 1$, then $y = y \cdot x \bmod p; x = x^2 \bmod p; i = i + 1$
- if $y \equiv c (\bmod p)$, accept. Otherwise, reject."

The algorithm runs in polynomial time. In the above algorithm, steps 1 and 4 will be executed once. The step 3 needs $O(n)$ time. Thus, *M* is a polynomial time algorithm for *MODEXP*.

Therefore, *MODEXP* $\in$ *P* .