```
Before solving the problems,
I'd love to note the definition for NP problem ahead.

A problem is in NP if there exists a NTM
that can verify a candidate solution in polynomial time.

Thus, for proving task, I think the main idea would be
figuring out the candidate  solution
and proving that the testing task would be done by NTM in poly-time.
```

Question 1

Call graphs G and H ***isomorphic*** if the nodes of G may be reordered so that it is identical to H. Let $ISO = \{\langle G, H\rangle |\ G \text{ and } H \text{ are isomorphic graphs}\}$. Show that $ISO \in \mathrm{NP}$.

To show $ISO \in \mathrm{NP}$:

1. **Candidate Solution**: A bijection $f : V(G) \to V(H)$ (a reordering of G's nodes to match H's nodes).
2. **Verification Process**:
   - **Non-deterministic Guess**: A non-deterministic Turing machine (NTM) guesses a bijection f in one step.
   - **Deterministic Check**: For every edge $(u, v) \in G$, check if $(f(u), f(v)) \in H$; for every non-edge $(u, v) \notin G$, check if $(f(u), f(v)) \notin H$. This check runs in polynomial time (proportional to the number of nodes and edges in G and H).

Since an NTM can verify ISO in polynomial time, $ISO \in \mathrm{NP}$.

Question 2

Show that if $\mathrm{P} = \mathrm{NP}$, you can factor integers in polynomial time.

1. **Establish that Integer Factorization is in NP**
   The integer factorization problem can be defined as: given a composite integer N, determine if there exist integers $a, k \geq 2$ such that $n = a \times k$.
   This problem belongs to NP because:

- For any candidate factors a, we can **verify** the solution in polynomial time by testing whether N % a == 0, which fits the definition of NP problem.

2. **Apply the Assumption $P = NP$**
   If we assume $P = NP$, then integer factorization is also in P.
3. **Conclusion**
   By definition of P, if $P = NP$, there exists a deterministic polynomial-time algorithm for integer factorization.