Here is the pumping lemma for regular languages:

If A is a regular language, then there is a number p (the pumping length) where, if s is any string in A of length at least p, then s may be divided into 3 pieces, s = xyz, satisfying the following conditions:

1. For each \( i \geq 0 \), \( xy^iz \in A \),
2. \(|y| > 0\), and
3. \(|xy| \leq p\)

1. Use the pumping lemma to prove that the following language is not regular:

\[ L = \{a^mb^na^k \mid k > m + n\} \]

**Proof by contradiction.**

If L is a regular language, then there is a number p where, if s is any string in L of length at least p, then s may be divided into 3 pieces, s=xyz, satisfying the three condition notes above. (*You need not include this, I just include it for completeness*)

Let \( s = a^pb^{p+2} \)

Note that \( s \in L \). Given condition 3, \(|xy| \leq p\), which means that y can contain only a’s from the start of the string (i.e., from the \( a^p \) part) and also y must contain at least one a because of condition 2. When we pump up, we therefore will add one or more a to the start of the string, so we will have: \( a^{p+x}b^{p+2} \), where \( x \geq 1 \). At this point \( k > m+n \) will not hold, since \( p+2 > p+x+1 \) does not hold, with \( x \) at least 1.

**Shortened version of proof that would get full credit:**

Let \( s = a^pb^{p+2} \). From condition 3, y must only contain a’s from the start of the string. When we pump up we add at least one additional a to the start of the string, since \(|y| > 0\), and hence the number of a’s at the end of the string will no longer be more than the number of a’s at the start of the string.