For #1 and #2, the problem asked for writing out the set (#1) or a clear English description of the language (both). Describing the operation — e.g. that LM means a string from L followed by a string from M — or regular expression — e.g. that $b(ab)^*$ is a b followed by one or more abs — only earned partial credit. What does LM look like for the particular L and M given in the problem? ($L = \{a, aa\}$ and M is strings ending with b, so LM would be all strings starting with a and ending with b. A b followed by one or more abs is strings of the form b, bab, $babab$, $bababab$, ... — alternating as and bs, starting and ending with b.)

Only use ... when the pattern is (simple and) clear, and even then, an English description can be preferable. For example: $\{\epsilon, a, aa, aaa, aaaa, \ldots\}$ is OK — the pattern can be easily seen as adding one more a to get the next string in the sequence. For something like "strings ending with b", however, the pattern is more complex because it needs to be able to express all strings of that form. At the very least, a methodical listing of set elements is required e.g. $\{b, ab, bb, aab, abb, bab, bbb, aaab, aabb, abab, abbb, baab, babb, bbab, bbbb, \ldots\}$ — but it takes more effort to recognize this as an enumeration of all combinations of a and b ending with b. (The English description "strings ending with b" is much easier.) Simply giving a few examples of things in the set doesn't work: $\{b, ab, aab, bab, bbb, abab, \ldots\}$ — what's the pattern?

Be careful to distinguish $\{\epsilon\}$ and $\{\emptyset\}$ from $\{\}$ and $\emptyset$ — $\{\}$ and $\emptyset$ denote the empty set (a set containing no elements), while $\{\epsilon\}$ and $\{\emptyset\}$ are both sets containing one element (the empty string and the empty set, respectively).

$\{ab^*\}$ technically is a set containing one string element ab^*. To express the language associated with the regular expression ab^*, one would properly write $L(ab^*)$ or $\{\ x \mid x \in L(ab^*)\ \}$ or $\{a, ab, abb, abbb, \ldots\}$ or $\{x \mid x$ consists of a followed by zero or more bs$\}$.

Be careful with precedence of operators: $ab(a|b|c)^*ba \mid aba$ means $L(ab(a|b|c)^*ba) \cup L(aba)$ — strings starting with ab and ending with ba. (aba doesn't need to be singled out in the English description because aba does start with ab and end with ba; it does need to be singled out in the regular expression because $ab(a|b|c)^*ba$ requires at least two bs, with possibly other characters between them.) To express "a string of as, bs, and cs, starting with ab and ending with either ba or aba, another set of parens is needed: $ab(a|b|c)^*(ba \mid aba)$. (As a side note, this would be more simply expressed as just $ab(a|b|c)^*ba$.)

"Contains empty strings" is a strange thing to say, since "contains" implies the presence of something and the empty string is characterized by the absence of characters. Describing $L(a^*(b|\epsilon)a^*)$ as "two sets of 0 or more as separated by a b or empty string" is strange; "two sets of 0 or more as optionally separated by a b" is better, but this is still awkward (how can a set of 0 things be separated by anything?) and also describes the regular expression more than the language. Recognizing that this set of strings is "strings with at most one b" is the best description of the language.

It was noted that the `/` symbols in a pattern like `([0-9]{1,2})/([0-9]{1,2})/([0-9]{4})` need to be escaped, but as this actually depends on the particular flavor of regex implementation, no points were taken off either way.