Error: Macro TranslatedPages(None) failed

```
expected string or buffer
```

AmigaDOS pattern matching

The use of wildcards in patterns allows you to set up very powerful searches. The special wildcard characters are listed below.

`?` Matches any single character

so Y?M matches YAM, YUM, Y@M etc.

`#` Matches zero or more occurrences of the following item

so Y#AM matches YM, YAM, YAAM, YAAAM etc.

`#?` Matches anything at all (including nothing!)

so #?YAM#? matches any string containing "YAM"

`%` Matches an empty string. Not terribly useful by itself, but you could use it to find messages with blank "Subject:" headers.

`()` Round brackets are used to group characters and expressions and show how to evaluate the expression. Use them freely!

Thus #(Re:)YAM matches YAM, Re:YAM, Re:Re:YAM ...

`[]` Square brackets are used to indicate a list of alternatives

so [YAM] matches any of the single letters Y, A or M but not YAM

`-` Indicates a range, so [0-9] matches any single digit and #[0-9] matches any integral number.

`~` Means NOT, referring to the whole expression following,

so ~Re: matches any string that does not start with "Re:"

`|` Means OR, referring to the expressions either side. You have to enclose the whole expression in round brackets:

e.g. (#?YAM#?|#?MUI#?) finds strings including either "YAM" or "MUI".

`'` Removes the meaning of a special character. For instance,

'#'? matches the literal string "#?"

WARNING! It's not as easy as it looks! For example, let's construct a filter that will find references to YAM or YAM2, but will disregard YAM1. Try this one: #?YAM(~1)

What does this say to do? Reading a new string from the left, look for the sequence **YAM**. If you find it, look at the following part: is it equal to "1"? If not, you've got a match. The trouble is, "1.3.5" isn't equal to "1" (it's too long),

so YAM1.3.5 is included though we don't want it. The solution is to make it clear that the string following "YAM" must not start with a "1". So try: #?YAM(~1#?)

This gets rid of 1.3.5 alright, but there's still a problem: sometimes, people insert a space between YAM and the version number and sometimes not. So references to **YAM 1** are still included. So we must say that any number of spaces may be present, like this: #?YAM# (~1#?)

Another case -- suppose we want to find all the strings that include **YAM** but do not start with "Re:". So we try: (~Re:)YAM#?

This says to begin by comparing the start of the string to "Re:". If we get a match, that string is discarded; if not, we look in the rest of the string to find "YAM". So what happens if the string starts with "YAM"? The first letter isn't **R**, so the NOT condition is satisfied. But we've done the Y now, so we don't find the string **YAM**! To sort this out, we have to explain that the string in front of "YAM" may be the null string (so that's what it is for!). Like this: (~Re:|%)YAM#?

Question for computer buffs: what does ~((~#?YAM#?)|(~#?MUI#?)) mean?