Question 1: *Down Pat*

A *pat* is a single letter or a string of letters which can be split into a left and right string (of at least 1 letter) where: each is the reverse of a *pat*; and all the letters in the left string are later in the alphabet than all the letters in the right string.

For example:

- BA is a pat as it splits into B and A, both of which are single letters and therefore pats, and B is alphabetically after A. AB is not a pat as the alphabetical rule would be broken;
- Similarly ED is a pat but DE is not;
- DEC is a pat as it splits into DE (whose reverse ED is a pat) and C.
- CEDAB splits into CED and AB, whose reverses are pats and C, E, and D are after A and B alphabetically.

1(a) [24 marks]

Write a program that reads in two strings from a line, s_1 then s_2, each between 1 and 6 *uppercase* letters inclusive.

You should output three lines, each containing a YES or NO indicating, in order, if s_1 is a pat, if s_2 is a pat, and if s_1s_2 (the combination of the two words) is a pat.

You must get all three lines of output correct to score marks.

Sample run

```
DE C
NO
YES
YES
```

1(b) [3 marks]

Which permutations of ABCD are pats?

1(c) [5 marks]

How many permutations of the alphabet, beginning with the letter B, are pats?