Bandit Level 11 - Level 12

The password is stored in data.txt where all lowercase and uppercase letters have beeen rotated by 13 positions.

Idea here is to move each letter, lower and uppercase 13 letters along. E.g. an 'a' or 'A' would become 'n' or 'N'. This rotation is called ROT13.

we will use the tr command. What we cannot do is simply map A to N and then O to B. If we think along the lines of reading the alphabet starting and A and ending in Z. You can map A to N, but not O to B, since B is the begining of the alphabet or the range does not exist / has passed.

The following command is

```
bandit11@bandit:~$ tr 'A-Za-z' 'N-ZA-Mn-za-m' < data.txt
The password is 7x16WNeHIi5YkIhWsfFIqoognUTyj9Q4
```

Lets break the tr 'A-Za-z' 'N-ZA-Mn-za-m' down. The first set of quotes is what is to be changed.

'A-Za-z'

In this case all uppercase and lowercase letters are to be included in the change / rotation. In other words the entire alphabet regardless of case.

The next set of quotes 'N-ZA-Mn-za-m' looks complicated at first, but if we break it down it makes sense.

'N-Z'. Maps uppecase letters to A to M to N to Z, the first half of the alphabet. So A becomes M. B becomes O and M becomes Z.

'A-M' Maps the remaining uppercase letters. So N becomes A, O becomes B and Z becomes M.

Note that we cannot simply write N-M since we have gone past the end which is Z. We have to break it down into two steps and ensure what we are within range.

The next step is exactly the same, except we are dealing with lowercase now.

'n-z' Maps lowercase letters a to m to n to z. So a becomes n, b becomes o and m becomes z.
'a-z' Maps the remaining lowercase letters, so n becomes a, o becomes b and m becomes z.

Finally the data.txt is what we are feeding into the tr command.