Avanan QA Planning Exercise

by Zeev Schneider

Description:

This is a planning exercise for the following:

Assume we are developing a code for "replace a string" software utility, activation is from the command line and it is activated as such that all occurrences of string2 within string1 will be replaced with string3:

./replace.py “string1” “string2” “string3”

for example:

./replace.py “dkmjeircuj874357mfdujm3eu934jjjj2398” “jjjj” “1984”

will result in the following output:

dkmjeircuj874357mfdujm3eu93419842398

Assumptions:

1. If not stated otherwise the following tests should be run in the same way on all 3 major OSs: MacOS, Windows, Linux
2. Pyton > 3.6 installed on the machine.
3. All example strings in this document appear without quotes – to save space
4. User should receive notification if something changed, and also a notification that nothing changed :

e.g

./replace.py “dkmjeircuj874357mfdujm3eu934jjjj2398” “jjjj” “1984”

will result in the following output:

Your input string was changed to the following: dkmjeircuj874357mfdujm3eu93419842398

./replace.py “Hello to you. Hello all” “hi” “1111”

Your input string was not changed – sub string not found. Returning original string: Hello to you. Hello all

1. All command line input is considered string:

e.g: “1234” and 1234 are strings – “”1234”” and “1234” accordingly.

Test Cases List:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | TestCase Description | String1 | String2 | String3 | Expected Output |
| 1. | Replace a single occurrence of a sub string. | dkmjeircuj874357mfdujm3eu934jjjj2398 | jjjj | 1984 | dkmjeircuj874357mfdujm3eu93419842398 |
| 2. | Replace multiple occurrences of a sub string | Hello sir, Hello madam, Hello all | Hello | Hi | Hi sir, Hi madam, Hi all |
| 3. | Replace occurrences both of upper and lower case | Hello sir, hello madam, hello all | hello | hi | hi sir, hi madam, hi all |
| 4. | Return original string if nothing found to replace | Hello sir, hello madam, hello all | sit | stand | Hello sir, hello madam, hello all |
| 5. | Except input of 1 char original string and replace or do not replace according to the rules above | G | g | 123 | 123 |
| G | 123 | J | G |
| 6. | Except a very large input of original string – 40K chars |  |  |  | Same replacement rules apply |
| 7. | Validate performance of the code in large inputs – original string, searched string and replacement string |  |  |  | The code should perform efficiently.  Note: Should receive definition for that from PM or Dev. |
| 8. | Test Unicode input for any or all the pieces | שלום לכם | שלום | Hello | לכם Hello |
| לכם Hello | Hello | שלום | שלום לכם |
| שלום לכם | שלום לכם | Hello to you | Hello to you |
| שלום לכם | שלום | היי | היי לכם |
| 9. | Ensure input of all parts of input – replacement string | Hello to you | hi |  | Hello to you  Note: You can replace a sub string by ‘’ |
| 10. | Ensure input of all parts of input – replacement string - 2 |  | Hello to you | hi | Hello to you  Note: Input is place related -so “Hello to you” is the original string “hi” is the substring and ‘’ is the replacement |
| 11. | Ensure input of all parts of input – substring | Hello to you |  | hi | Hello to you  Note: Input is place related -so “Hello to you” is the original string “hi” is the substring and ‘’ is the replacement |
| 12. | Ensure input of all parts of input – substring 2 | Hello to you |  |  | System error: No substring was provided |
| 13. | Ensure no recursion in the code | Jjj12123434jj | 1234 |  | Jjj1234jj |