Question 3 [50 points]

The goal of this task is to write functions to identify certain repeated patterns of characters in long character vectors, a basic form of a more complicated task that is often used in gene sequencing.

For every part of this question, you will assume that the user gives you a vector where each element of the vector contains a single character,lower-case letter. For example, the user may specify:

```
c("b", "c", "b", "d", "c", "a", "b", "b", "d", "c")
```

(a) **[15 pts]** Write a function below using a `for` loop (and possibly other control statements) which takes a character vector as an argument and returns the length of the longest sequence of repeated letter **"b"** for an arbitrary vector. For the example vector above, for example, the length of the longest sequence of repeated **"b"** values is `2`. It does not matter if the longest sequence length occurs multiple times, you only need to report it once.

Answer:

CONTINUED ON NEXT PAGE

(b) [**15 pts**] Now assume that if the user inputs a vector that includes a certain stopping character, then you should immediately **stop** analyzing the sequence and return a value of `NA`. If the input vector does not include the stopping character, then it proceeds as in part (a) to return the length of the longest sequence of repeated letter **"b"** values. For example, if the stopping character is "a", then in the example above, your function should return `NA`. But if the stopping character is "f", then in the example above should return `2` as before. Modify your function from part (a) to complete this task. Your function should take two arguments: the input character vector and a stopping character whose default value is "f".

Answer:

CONTINUED ON NEXT PAGE

(c) [**10 pts**] Now assume that you want write code to create a data frame or tibble that contains the longest run in the vector for each letter of the alphabet, **except** for the single special stopping character specified by the user. If a non-stopping letter does NOT appear in the vector, it should not appear in the table. In other words, if the stopping character is "f", then applying your code to the example vector above would return.

```
# A tibble: 4 x 2
  letter longest
  <chr>    <dbl>
1 a            1
2 b            2
3 c            1
4 d            1
```

But if the stopping character is "a", then your function should return `NA` for all letters, i.e.

```
# A tibble: 4 x 2
  letter longest
  <chr>  <lgl>
1 a      NA
2 b      NA
3 c      NA
4 d      NA
```

Write code below that uses your function from part (b) to produce the desired result. You do not need to write a separate function for this part, but you can if you think it is helpful.

Answer:

CONTINUED ON NEXT PAGE

(d) [**10 pts**] Finally, using your code from part (c) so that you can obtain a **list** with 26 elements, where you obtain the tibble in part (c) for a each of the 26 possible stopping characters. You do not need to write a separate function for this part, but you can if you think it is helpful.

END OF QUESTION