Kyle Connolly, kc5492@rit.edu
Vedika Vishwanath Painjane, vp2312@rit.edu
Kushal kale , ksk7657@rit.edu

1. **Linear search**

```
def findLastIndex(L: list, prefix: str, start_i: int) -> int:
    if start_i == None:
    return -1
    for i, word in enumerate(L[start_i:]):
        If not has_prefix(word, prefix):
            return i - 1 + start_i
    return -1 # case where prefix does not exist
```

2. **Words starting with “bb”, “a”, “d”**
   a. “bb”, 5, 5
   b. “a”, 0, 0
   c. “d”, None, 10
3. **Changes to binary search**
   On line 7, instead of matching the whole word, we need to check if the word has the prefix(val)
   has_prefix(data[midindex], val)

4. **Function printing user input, sentinel value considered**

```
def user_input():
    while (word := input(‘Please enter word: ‘)) != "<QUIT>":
        print(word)
```

Bonus - hasPrefix function

```
def hasPrefix(word):
```

Recursive call midindex - 1
else has_prefix is not equal to val
midindex+1