LeetCode 271 [Encode and Decode Strings]

Design an algorithm to encode a list of strings to a string. The encoded string is then sent over the network and is decoded back to the original list of strings.

s: [ab, a, abc, d]

es: [xxxxxxx]

ds: [ab, a, abc, d]

Approach 1: use extra character

s: [ab, b, c, abc]

es: [ab#|b#|c#|abc]

ds: [ab, b, c, abc]

Time: O(n)
Space: O(1)

Java solution

```
public String encode(List<String> strs) {
    if (strs.isEmpty()) {
        return Character.toString((char) 258);
    }

    String separate = Character.toString((char) 257);

    StringBuilder sb = new StringBuilder();
    for (String s: strs) {
        sb.append(s);
        sb.append(separate);
    }
    sb.deleteCharAt(sb.length()-1);

    return sb.toString();
}

public List<String> decode(String s) {
    if (s.equals(Character.toString((char) 258))) {
        return new ArrayList<>();
    }
    String separate = Character.toString((char) 257);
    return Arrays.asList(s.split(separate, -1));
}
```

Kotlin solution

```
Fun encode (strs: List<String>): String {
    if (strs.isEmpty()) return 258.toChar().toString()

    val separate = 257.toChar().toString()

    val sb = StringBuilder()
    for(s in strs) {
        sb.append(s)
        sb.append(separate)
    }
    sb.delete(charAt(sb.length -1)

    return sb.toString()
}

Fun decode (s: String): List<String> {
    if (s == 258.toChar().toString()) return emptyList()

    val separate = 257.toChar().toString()

    return s.split(separate)
}
```