Question 1

Your colleague is implementing a new feature which fetches all models of a particular car brand from a web API then displays them on screen. For example, the API will return *A*, *B*, *C* and the other models for an example brand. Your colleague has requested you as a reviewer to review the below implementation. Therefore, please provide ALL comments for where you think could be applied including architecture and coding conventions. Feel free to add classes and move code around if you think comment is not enough to elaborate your thought.

```
interface ApiClient {

    fun fetchModels(): Flow<List<String>>
}
```

```
// Use Koin for dependency injection
val myModule = module {
    factory { ApiClient() }
}
```

```
class MainActivity : ComponentActivity() {
    var apiClient: ApiClient = get()
    var msf: MutableStateFlow<List<String>> = MutableStateFlow(emptyList())

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            MyApplicationTheme {
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    showModelList()
                }
            }
        }

        GlobalScope.launch {
            apiClient
                .fetchModels()
                .flowOn(Dispatchers.IO)
                .catch {
                    Log.e("MainViewModel", "Failed to get list of car models.")
                }
                .collect {
                    msf.value = it
                }
        }
    }

    @Composable
    fun showModelList(modifier: Modifier = Modifier) {
        val l = msf.collectAsState().value
        LazyColumn(Modifier.fillMaxSize()) {
            items(l) {
                ListItem(it)
            }
        }
    }

    @Composable
    fun ListItem(item: String, modifier: Modifier = Modifier) {
        Row(modifier.fillMaxWidth(), horizontalArrangement =
Arrangement.Center) {
            Text(text = item)
        }
    }
}
```

Question 2

In any OS, it is essential to be able to sort files in a directory according to their names. Now, you're required to implement such a sort function which could arrange the following list of input files to the expected output.

Input:

```
file2.gif
file01.gif
1file.jpg
1file.gif
file10.gif
file1.gif
file1a.gif
```

Output:

```
1file.gif
1file.jpg
file1.gif
file01.gif
file1a.gif
file2.gif
file10.gif
```

Please complete the function below which should implement the desired sorting. You're allowed to use any built-in classes in Java or Kotlin and their functions.

As a hint, you may start from the basic input and output.

Input:

```
1file.png
file10.gif
file2.gif
```

Output:

```
1file.png
file2.gif
file10.gif
```

```java
class Solution {
    public String[] sort(String[] fileNames) {

    }
}
```

```kotlin
class Solution {
    fun sort(fileNames: Array<String>): Array<String> {

    }
}
```

Note:

- Only letters and numbers can be used in a file name.
- `1 <= fileNames.length <= 100`
- `1 <= fileNames[i].length <= 100`