Cairo Crash Course

Advanced Data Types - Exercise 6

In this exercise we will will work with the `List<T>` trait from Alexandria library.

Contract Implementation

In `advanced_types_6.cairo` implement the contract `SimpleList` according to the following instructions:

1. Define the following contract Storage

```
odd_numbers_list:  List<u32>,
even_numbers_list: List<u32>,
```

2. Implement the public function defined in the interface:

```
 ISimpleList<TContractState> {
    fn odd_or_even(ref self: TContractState, number: u32);
    fn get_number_by_index(self: @TContractState, list_name: felt252, index:u32) -> u32;
}
```

- `odd_or_even()` - Receives a number and writes it to storage. If the number is even, it should be appended to `even_numbers_list`, otherwise it should be appended to `odd_numbers_list`.
- `get_number_by_index()` - Receives a list name and an index, and returns the number stored in the list at the given index.

Writing Tests

In `test_advanced_types_6.cairo`, inside the function `test_simple_list()`, follow the next steps to complete the tests:

1. Deploy the `SimpleList` contract and create a dispatcher.
2. Test the functionality according to the TODOs.