```python
import sys

from pipelines.registry_pipelines import single_registry_pipeline, all_registries_pipeline
from utils.fms import init_registry_dirs, get_registry_codes
from utils.helpers import handle_cli_verbosity


def test_single_registry(argv):
    """
    Runs steps to extract aircraft registry test from a specific remote source, transform the extracted test into a
    standardized format, and load the transformed test by integrating it into tabular stores used to seed or update
    components of a data warehouse

    :param (list) argv: command line arguments - pass registry-code for the target remote source to update;
        pass test size to sample from each aircraft registry data source;
        pass 'verbose' to print progress messages
    :return: effect - creates [TEST_DIR]/world[REGISTRY.name] file;
        effect - creates [TEST_DIR]/aircrafts[TABLE.name] file;
        effect - creates [TEST_DIR]/registrants[TABLE.name] file;
        effect - creates [TEST_DIR]/registrations[TABLE.name] file;
        effect - creates [TEST_DIR]/[REGISTRY_INPUT_DIR]/[argv[1]]/[file or directory] asset(s);
        effect - creates [TEST_DIR]/[REGISTRY_OUTPUT_DIR]/[argv[1]]/[argv[1]][REGISTRY.name] file
    """

    # initialize registry test directories
    init_registry_dirs(True)

    if len(argv) <= 1:
        # handle when registry-code of remote source to process is missing
        sys.exit("Error: must pass registry-code of remote source to process")

    # assign registry-code of remote source to process
    registry_code = argv[1].lower()

    if registry_code.upper() not in get_registry_codes():
        # handle when registry-code of remote source to process is invalid
        sys.exit("Error: invalid registry-code of remote source to process")

    if len(argv) <= 2:
        # handle when test size is missing
        sys.exit("Error: must pass test size to sample each aircraft registry data source by")

    if not argv[2].isdigit():
        # handle when test size is invalid
        sys.exit("Error: invalid test size to sample each aircraft registry data source by")

    # assign test size
    test_size = int(argv[2])

    # assign whether progress messages should be printed
    verbose = handle_cli_verbosity(argv, 3)

    # extract, transform, and load remote registry data source
    single_registry_pipeline(registry_code, verbose=verbose, is_test=True, test_size=test_size)
    # compile world registry data
    all_registries_pipeline(verbose, read_only=True, is_test=True)


if __name__ == "__main__":
    test_single_registry(sys.argv)
```