```python
import sys

from pipelines.registry_pipelines import all_registries_pipeline
from utils.fms import init_registry_dirs
from utils.helpers import handle_cli_verbosity


def test_all_registries(argv):
    """
    Runs steps to extract aircraft registry tests from all remote sources, transform the extracted tests into a
    standardized format, and load the transformed tests into tabular stores used to seed or update components of a data
    warehouse

    :param (list) argv: command line arguments - 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]/[registry_code]/[file or directory] asset(s) for each remote source;
        effect - creates [TEST_DIR]/[REGISTRY_OUTPUT_DIR]/[registry_code]/[registry_code][REGISTRY.name] file for each remote source
    """

    # initialize registry test directories
    init_registry_dirs(True)

    if len(argv) <= 1:
        # handle when test size is missing
        sys.exit("Error: must pass test size to sample each aircraft registry data source by")

    if not argv[1].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[1])

    # assign whether progress messages should be printed
    verbose = handle_cli_verbosity(argv, 2)

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


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