# Documentation of Testing

It is important that tests are described before they are carried out. The description of every test case should include

- the date of the test
- the name of the tester
- the identity of the component under test
- the purpose of the test
- the conditions of the test
- the input data
- the output data
- the expected results
- the actual results
- Tests should cover valid and invalid data
- Each test should have a known answer before you run it

and contain sufficient detail for anyone to duplicate later. Do not experiment, always know the expected results before testing. The description of test-cases should start as a module is being designed when the programmer will be most aware of possible trouble spots.

Testing is not carried out in a 'trial and error' manner. It must be logically planned as a set of test cases each of which are designed to examine some particular facet of the software. The results of each test must be evaluated with the expected results, and if unsuccessful will lead to debugging else to the next test case.

Your documentation of testing will contain two different sections:-
1 the testing strategy i.e. the method used to build the program while at the same time conducting tests to uncover defects.

2 testing the completed program:-
- the test-cases you carried out to ensure correct performance as an integrated system: that it functions as specified and produces correct results as functioned. That is, an exhaustive range of black-box tests, and a few white-box tests designed to check problematic areas.



# Documentation of Testing

Tests to perform :

1. Check file loads
2. Check data can be saved back to file
3. Check sorted data is sorted.
4. Check the number of items is the same before and after sorting.
5. Check results
6. Check updated table
7. Check item selected is correct
8. Load empty file
9. Load a file that does not exist
10. Check data type on file
11. Any other relevant test