

Chapter 4: More Test Cases, Experiences

Our group now has 25 test cases (in *exampleTests.txt* in our GitHub repository) for the *makeplural()* and *makesingular()* functions in *objnam.c*. All test cases are testing independent portions of the code, as they should, making them unique and non-redundant.

The framework is built and outputs: tested-output correctness, function being tested, it's arguments and expected output, as well as the total time elapsed on completion of the test.

Example test cases. All test case lines must begin with a "\$". Other lines are assumed to be comments.

Test case format is "\$ functionname arg1 [...] argN expectedOutput".

First 3 examples show how greater detail surrounding each case can be achieved as desired.

```
#####  
#####  
## makeplural tests: makeplural receives a singular word and returns its plural  
counterpart ##  
#####  
#####
```

Test case 1:

```
$ makeplural human humans
```

Base case: PASS assures basic consonant-s format sing-to-plur conversion.

Test case 2:

```
$ makeplural valkyrie valkyries
```

Base case: PASS assures basic vowel('e')-s format sing-to-plur conversion.

Test case 3:

```
$ makeplural man men
```

Badman: PASS assures basic -man to -men sing-to-plur conversion.

```
$ makeplural homunculus homunculi
```

```
-us to -i
```

```
$ makeplural larva larvae
```

```
-a to -ae
```

```
$ makeplural wolf wolves
```

```
-f to -ves
```

```
$ makeplural nerf nerfs
```

```
-f to -ves: avoiding nerf -> nerves
```

```
$ makeplural mycelium mycelia  
-ium to -ia
```

```
$ makeplural nemesis nemeses  
-sis to -ses
```

```
$ makeplural box boxes  
-z,-s,-x to -zes, -ses, -xes
```

```
$ makeplural ruby rubies  
-y to -ies
```

```
$ makeplural matzo matzot  
-zo to -zot
```

```
$ makeplural knife knives  
-fe to -ves
```

```
#####  
#####  
## makesingular tests: makesingular receives a plural word and returns its  
singular counterpart ##  
#####  
#####
```

```
$ makesingular pigs          pig  
$ makesingular valkyries    valkyrie  
$ makesingular lizardmen    lizardman  
$ makesingular fungi        fungus  
$ makesingular algae        alga  
$ makesingular wolves       wolf  
$ makesingular balactheria  balactherium  
$ makesingular nemeses      nemesis  
$ makesingular boxes        box  
$ makesingular rubies       ruby  
$ makesingular matzot       matzo  
$ makesingular knives       knife
```