

Making a plan to fail

How we did it:

For adding the faults in the code, we created "add_faults.sh" and "remove_faults.sh". These two files swap out our code with the faulty code in a quick and simple command './add_faults.sh' or './remove_faults.sh' (see screenshot at the bottom for example execution)

We made the following modifications to the files EnDe.js, EnDeCheck.js, EnDeText.js, and what is swapped with the 'add_faults.sh':

Test 10: Modified the reverse function in EnDe.js to only go to length - 1

```
Changed line from:
var i = src.length;

Changed line to:
var i = src.length - 1;
```

Test 11: Modified the val2num function in EnDeCheck.js to replace non-digit characters with '-' instead of ""

```
Changed line from:
this.val2num = function(src) { return((src+='').replace(/[^0-9]/g, '')); };

Changed line to:
this.val2num = function(src) { return((src+='').replace(/[^0-9]/g, '-')); };
```

Test 12: Modified the atbash function in EnDe.js to change from -1 to -2

```
Changed line from:
bux += String.fromCharCode((((78-ccc)*2)-1+ccc));

Changed line to:
bux += String.fromCharCode((((78-ccc)*2)-2+ccc));
```

Test 15: Modified the DELwhite function in EnDeText.js to replace white space with '.' instead of ""

```
Changed line from:
case 'txtDELwhite': bux = bux.replace(/[\t \r\n]/g, ''); break;

Changed Line to:
case 'txtDELwhite': bux = bux.replace(/[\t \r\n]/g, '.'); break;
```

Test 18: Modified the stibitz function in EnDe.js to have incorrect value for case 1 from '0100' to '1111'

```
Changed line from:
case '1' : bux += '0100' + delimiter; break;

Changed line to:
case '1' : bux += '1111' + delimiter; break;
```

The Sample Run:

These fault changes were saved in `EnDe[Check|Test].js.fault`, and the preserved, passing versions were saved in `EnDe[Check|Test].js.normal` and the two sets of versions are copied in and out of the actual files run, `EnDe[Check|Test].js`. These operations were combined into the two files `add_faults.sh` and `remove_faults.sh` which allow for easy switching between the two states.

Update to the original code as well

As a side note, we have updated and 'streamlined' our code. Now, each test case runs as 'its own executable'. This allows us to test individual cases over running the entire test each time for one update. For instance 'python ./scripts/runAllTests.py 10 11 12 15 18' will only run those 5 tests. (see previous screenshot for sample execution of individual tests) Also updated the reports so that the chart contains 'pass' and 'fail'.

Screenshot of individual tests (faluts):

```
eze@ubuntu:~/Desktop/TeamToo-scratch/TestAutomation$ ./add_faults.sh
eze@ubuntu:~/Desktop/TeamToo-scratch/TestAutomation$ python scripts/runAllTests.py
py 10 11 12 15 18
Running Test #10...
FAILED
done
Running Test #11...
FAILED
done
Running Test #12...
FAILED
done
Running Test #15...
FAILED
done
Running Test #18...
FAILED
done
eze@ubuntu:~/Desktop/TeamToo-scratch/TestAutomation$
```

Screenshot of running all:

```
eze@ubuntu:~$ cd '/home/eze/Desktop/TeamToo-scratch/TestAutomation'
eze@ubuntu:~/Desktop/TeamToo-scratch/TestAutomation$ python scripts/runAllTests.py
Running Test #1...
Base64
PASSED
done
Running Test #2...
Hex
PASSED
done
Running Test #3...
integer to binary
PASSED
done
Running Test #4...
Rot13
PASSED
done
Running Test #5...
Morse
FAILED
done
Running Test #6...
PASSED
done
Running Test #7...
PASSED
```