

## “Brief Description of how you tested your structures”

### Queue:

The testing process was very simple really. I would use the *qwerty* testing method. I would enter q <hit enter> w <hit enter> etc and so on until qwerty had been entered. Then I would enter a period . <hit enter>. This would then print out to me the results. I would expect a perfect result of q w e r t y spelled back to me in order. In addition, I tried with more than one letter on a line and it will choose the first letter.

#### Queue test 1

```
q
w
e
r
t
y
.
Printing and removing the Queue.
q w e r t y
Queue is now empty.
flip3 ~/CS162/6week/mod2 163% █
```

#### Queue test 2

```
qw
we
er
rt
ty
yu
.
Printing and removing the Queue.
q w e r t y
Queue is now empty.
flip3 ~/CS162/6week/mod2 164% █
```

## Stack

For stack, the process was the same. IE I would use the *qwerty* testing method again. On the stack, I would expect a perfect result of *y t r e w q* spelled back to me in backwards order. Like before, I tried with more than one letter on a line and it will choose the first letter like in test 2.

### Stack Test 1

```
q
w
e
r
t
y
.
Printing and removing the Stack.
y t r e w q Stack is empty.
flip3 ~/CS162/6week/mod2 162% █
```

### Stack Test 2

```
qw
we
er
rt
ty
yu
.
Printing and removing the Stack.
y t r e w q Stack is empty.
flip3 ~/CS162/6week/mod2 167% █
```

On both Stack and Queue, I knew I had done it correct if the results came back in perfect expected order.