

3P95 A2 Part 1:

Code 2:

There is only one predicate, $n=0$. Testing was done on 10 values spanning -5 to 4. These will be represented as -1, 0, and 1.

	-1	0	1	Failure	Context	Increase
$n==0$	0	1	*	0.33	1	-0.66
Outcome	F	S	S			

The bug found was that any value less than 0 input into the system caused a stack overflow error. This is because it kept trying to return $n * \text{factorial}(n-1)$, but $\text{factorial}(\text{int } n)$ terminates on $n=0$. This means that any negative numbers processed will result in a stack overflow error. I found this error by running the code from $n=-5$ to $n=4$, all the inputs <0 produced a stack overflow.

I have also determined the other two bugs.

Code snippet 1: Binary search

This was supplied with an ordered array, incrementing by a certain ratio from each cell to the next. This was done with 1,2,3,4... , 0,2,4,6,8... and 0,3,6,9...

Results found that it succeeded for values within $[1, \text{length}]$ for the first array. However, it failed for any values beyond that. It also failed with any values not in the second and third arrays. Further testing has shown that if the value falls between two values of the array or the value is less than the smallest value, it fails. This means that the error must be when the bottom half of the array is searched, and this proved true. The function should be:

```
Return binarySearch(arr,l,mid-1,x);
```

But is instead

```
Return binarySearch(arr,l,mid,x);
```

This causes an infinite loop.

Code snippet 3: Bubble sort

If this is supposed to be a range limited bubble sort, then there are numerous problems. If not, then this fails because if any value is greater than the limit value, the sort stops and the array is returned. This was tested with a sequence of random arrays of increasing sizes, every time a value was greater than the limit value, the process was ended prematurely.