

[3.25] Internal Error with seq() and Boolean operators in list index calculation

TI confirms that list index calculations involving a Boolean operator will fail with an Internal Error, when the calculation is performed in a `seq()` function call. This limitation applies to AMS 2.05. This will fail:

```
seq(t[1 and 0hF],k,0,1)
```

In this example, *t* is the list, and the Boolean *and* operator is used to calculate the index. *k* is the sequence variable, and its limits are immaterial. Remember that Boolean operations on integers must be executed in Exact or Auto mode; in Approx mode they will fail with a Data Type error.

The last example is the simplest possible command which demonstrates the error, but you would not likely attempt this, anyway. A more realistic example which fails is

```
seq(t[(shift(n,k*-4) and 0hF)],k,0,7)
```

Again, the offending party is the Boolean *and* operator. Any of the Boolean operators (*and*, *or*, *not*, *xor*) will cause the error. The index calculation itself does not cause the error, for example, this works:

```
0→k  
t[(shift(n,k*-4) and 0hF)+1]
```

A Boolean operator will also cause an error if the summation function is used:

```
Σ(t[(shift(n,-4*k) and 0hF)+1],k,0,7)
```

There may be other built-in functions which are also susceptible to this error.

In most cases you can work around this limitation. I found the limitation while attempting this command:

```
sum(seq(t[(shift(n,k*-4) and 0hF)+1],k,0,7))→s
```

The equivalent work-around is

```
0→s  
for k,1,8  
  s+t[(n and 0hF)+1]→s  
  shift(n,-4)→n  
endfor
```

The work-around is effective because the index calculation is not embedded in a `seq()` function call.

I emailed TI Cares about this, and they claim it will be fixed in the next AMS revision.