

Last login: Wed Mar 22 13:15:59 on ttys006
carbon:SamplePrograms\$ utop

Welcome to utop version 1.14 (using OCaml version 4.01.0)!

Type #utop_help for help about using utop.

```
-( 18:00:00 )-< command 0 >-----{ counter: 0 }-
utop # let cond c t f = if c then t else f
;;
val cond : bool -> 'a -> 'a -> 'a = <fun>
-( 13:48:11 )-< command 1 >-----{ counter: 0 }-
utop # let rec sumToN n = cond (n = 0) 0 (n + sumToN (n-1)) ;;
val sumToN : int -> int = <fun>
-( 13:48:20 )-< command 2 >-----{ counter: 0 }-
utop # sumToN 5;;
Stack overflow during evaluation (looping recursion?).
-( 13:49:05 )-< command 3 >-----{ counter: 0 }-
utop # let rec sumToN' n = if(n = 0) then 0 else (n + sumToN' (n-1)) ;;
val sumToN' : int -> int = <fun>
-( 13:49:13 )-< command 4 >-----{ counter: 0 }-
utop # sumToN' 5;;
Stack overflow during evaluation (looping recursion?).
-( 13:49:33 )-< command 5 >-----{ counter: 0 }-
utop # let rec sumToN' n = if(n = 0) then 0 else (n + sumToN' (n-1)) ;;
val sumToN' : int -> int = <fun>
-( 13:49:36 )-< command 6 >-----{ counter: 0 }-
utop # sumToN' 5;;
- : int = 15
-( 13:49:46 )-< command 7 >-----{ counter: 0 }-
utop # sumToN ;;
- : int -> int = <fun>
-( 13:49:48 )-< command 8 >-----{ counter: 0 }-
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Call
-----	--------------	-------	-------------	----------------	---------	----------	--------	------