	Representation (bits)			Representation	ntation
	31-bit, 2's comp	olement number	tag bit (1=num, 0=bool)	hex	decimal
9	0000 0000	0 0000 0000 0000 0	000 000 <b>1 0011</b>	0x00000013	19
-2	1111 1111	1111 1111 1111 1111 1111 1111 1111 <b>1101</b>		0xFFFFFF <b>D</b>	-3
33	0000 0000	0000 0000 0000 0000 0000		0x000000	
true	<b>1</b> 111 1111	<b>1</b> 111 1111 1111 1111 1111 1111 1110		0x <b>F</b> FFFFF <b>E</b>	-2
false	<b>0</b> 111 1111	1111 1111 1111 1	111 1111 111 <b>0</b>	0x <b>7</b> FFFFFF <b>E</b>	214
	true/false	30 bits for f	 uture value representation	s!	
EBoo	l(b) (* b is t	rue or false *	->		
	esp-4, which contai	•	(Assume x at esp-		•
	esp-4, which contai /hat if esp-4 contaiı	•	(Assume x at esp- (Assume y at esp-		
	•	•	•	8, which contains	
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD
(Follow-up: W	•	ned 0x7FFFFFE?)	(Assume y at esp-	8, which contains	0xFFFFFFD

If the contents of EAX are less than or equal to the contents of EBX, jump to the label done.

or arg1, arg2 — bitwise or the args, store in arg1

jle done

(Assume x at esp-4, which	contains 0x0000013)	(Assume x at esp-4, which contains 0x0000013)  (Assume y at esp-8, which contains 0xFFFFFFD)		
(Follow-up: What if esp-4	contained 0x7FFFFFE?)			
(add1 x)	In EAX at end?	(> x y)	In EAX at end?	
Example  cmp eax, ebx  jle done		arg2 – bitwise and th		

If the contents of EAX are less than or equal to the contents of EBX, jump to the label done.

```
Value
                         Representation (bits)
                                                                    Representation
                                                                            decimal
                                                                 hex
             31-bit, 2's complement number
                                          tag bit (1=num, 0=bool)
     9
                 0000 0000 0000 0000 0000 0000 0001 0011
                                                              0x00000013
                                                                             19
     -2
                 1111 1111 1111 1111 1111 1111 1111 1101
                                                              0xfffffffD
                                                                             -3
                 0000 0000 0000 0000 0000 0000 _____
    33
                                                              0x000000
   true
                 1111 1111 1111 1111 1111 1111 1110
                                                                             -2
                                                              0xFFFFFFE
   false
                 214...
                                                              0x7FFFFFE
             true/false
                               30 bits for future value representations!
let rec e to is (e : expr) (si : int) (env : tenv) =
  match e with
    | ENum(n) ->
    | EBool(b) (* b is true or false *) ->
```