15-2 12 = (w w) $w = (\lambda x, xx)$ bytecode = bytecode C= < w, 0> CE, O, SL, mt7 [] = [w] Lw] ap H> CE, O, wwap, m+> $[Lw] = ([X, x \times ap)]$ > < < w, d7, d, wap, m+> = < c, ø, wap, m+> vseless (1x, xxap) (-1x, xxap) ap < C C C , Ø , ap , m+> / // H> < E, [XH>C], (XX), < E, Ø, 2, m+>> -> < E, [XHO], XXAP, A> HOCC, [XHOC], X ap, A> HOCCI, [XHO], ap, FT> H) (E, [XH), (XX), (E, [XH)], E, A) useless CEK allocates on argevalvation, frees on fin calls SECD allocs on calls and arg eval "O(a) 0(a+c) Trist (FF) 0) Vntrust == (Af, Dx. ((CF) (1x.x))) CTRUST, CUNTRUST (TRUST?) < (MN), E, KD FOR < M, E, < ar, N, E, K>> < (JXIM), E, K) +> < < (JXIM), E>, E, K> E|s = E exempt s (restricting E to S) 1 safe 2x1,4123/8x3 = 8x123 for Space " () H < M, E FU(m), < ar, N, E FV(N), K>> (2) H) < < (UKim), El Fr/m) 7, Ø, K) prevents sharing (c.f. "flat closures")

6-1) Does ISWIM have errors? Enothing - stuck M +> [Enothing
- (x 7) (unbound variables are stick)
(5 7)
(110) 8 is partial
(+ (\(\lambda(x)x)\)S) & takes only b
evalu ((57)) =? nothing = evalu (12)
pantal
evalv (M) = em if M=>>N, and Nis stock
Error-ISWIM
M = , , lerr, 1 & Labels (unbound, not-a-form, imalib-arg, , ,) Epartial
$Old: S: O^n \times b^n \rightarrow V$ (pantial)
New: 8: on x b" -> V v err, (total)
S(V, 21,07) = errdivbyzero
8(1, 62,17) = 2
& (on bin (-1 (x) M) bin) -> errnot constant
$(b V) \rightarrow evrnotation \qquad m = (0^n m)$
$(+123)$ 0 > errormanyargs $= (6n m_{111}n)$
3: not a program
(x5) 0 = errunbound (2) not a program
(+ (110)5) => (+ erraino 5)
-> errnot constant
$((\lambda(x) (+ x S)) (1 10)) \rightarrow (\dots errino) \rightarrow (+ divo S) \rightarrow$
em constant

	(x) 5) (/(0))
	1(x) 5) erroino) similar to laziness 5 //not C by Haskell
	5 // NoT C by Flaskell
Jelf-kriegenighter, og med grek den finne til frædtik men ett til enen en _{ene} og enen stresse kriene skrive skrive skrive skrive skrive.	
Inspired and to end on a fine and major and a second and	(on V err, M) -> err,
	(err, M) -> err,
migrafialanti (1) Migrafia displicati di più liggi di misori di diri sidranjo, coppe a grappi ny actione in cin 10 Mil 10	(V erri) -> erri
santembellas ann reason rédente de anno qui étiquir en casa anal, casa e qualitar en coment mille à mil	(V, (or Vz (V3 erri) M)) => erri
	E = CJ (EM) (VE) $(on V EM)$
non-silven alternal access cities before per of discount a service colonic access parameters of the per of the self-	E[err,] -> err,
min to color finally community against a distribution of the color for t	printout E Ka Wall
	Doom
	$E[M] \rightarrow E[N]$
and to where the required in the properties of t	$E[((\lambda(x)m)v)] \longrightarrow E[mcx \leftarrow v]]$
sistence active the different active and active act	What's the connection between ISWIM and Err-ISWIM?
	Extension Theorem:
	For any M in ISWIM,
	1. $evalv(m) = A = evale(m) = A$
	2. evale (m) = A and A terr, => evalv(m)=A
	3. evale (m) = err, => evalv (m) is undefined
	Y, evale (m) is undefined => evalu (m) is undefined
	M = N , FF \ \ C. C[m] = C[N]
	M = N 'th AC' CTM? - CTM?
	? ~ = ~ ?
	$\frac{2}{V} \Rightarrow \frac{2}{V}e$
	~ = > ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
the region of the contract of	

16-> For all M and N M ISWIM, MZeN=7 MZVN ? TRUE The and easy MZVN =7 MZeN ? FALSE distinguishes identifies in Finite 1000 errors and loops with error 7 m, N 20 s, t. (+ C, C[N] = C[N]) (FC, C[m] te C[N]) W= (7t, (1x, ((+x) 12))) N= (Af. (Ax, D)) $C = ((L] (\lambda_x, err,)) (\lambda_{x,x})$ ([m] = err, C[W] = liverge Module A exports f f takes a function argument call f with a function that's errors (g) re C rv expressiveness observations

8,10 is interesting theorem