



Caefine (Yg)
(let (LS (lamboda (u)

Y Cg):

s = 2 -> g (26(x)) s(s)

G(f): (n) -> if n==0 then 1 else n \* f (n-1)

Y (g):

s(s)
where s = (n) -> g(n(x))

•

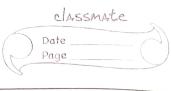
٧(٢) = ٥(٤)

= 9 (8 (S)) • 9 (9 (8 (S)))

- g (g (\*(\$)))

Ya (g): s(s)

where  $s = (x) \rightarrow (n) \rightarrow g(x(x))(n)$ 



	Faye
	ases syntax define-datatype
	Cond number? null? list? symbol?
	(car) (birst second) (memq) (adr) (cons) (adr)??
	(match) (match-lambda). (match-let).
	Cdefine-datatype ast ast?
	[num (n number?)]
	[plus (left mast?) (right ast?)]
	Eminus (left ast?) (night ast?)]
	[mul Cleft act?) (night ast?)])
	(2 ca)
	Crases ast a
	[num (n) n]
	[plus Clebt regint) (+ Negt majnit) (1)
	CEL (eval left)]
	[r (eval right)]) (+ l r))]
	Eminus Cleft right)
-	(tet
	(El (eval left)]
	[r (eval right)]) (+ e r))]
	(† (r)) <u> </u>

	classmate	8
	Date	
1	Page	

Cmemq v	give l	ist) ->	index		
(apply).	<b>#</b> †	#4 (1	malaam?		
 (apply).		40	200(201)		
,					
		· · · · · · · · · · · · · · · · · · ·			
		-		·	
				. ^	