hop declaration rules:
1) Syntax or the lam is
run [[([[varrame] = [integer])([]varrame]conditional operator [] integer)
space [warmanefincrement/decrement])][] { /n }
2) Spaces should not be missed else it would be treated as
Invalid Syntax.
Start with & and end with & and breeze prettores to appear can be any alphanumeric). Yun ((abc) invalid variable name * *abc+ !
4) There is no need to give Data Type begove var declaration. run (Point * abc* = 1) X
it is considered to be nopoint by default.
5) Variable must be the initialized in <inializers (="" *ab*)="" allowed.<="" declaration="" initiation="" is="" no="" not="" only="" part.="" point="" run="" td=""></inializers>
c) run (point *abc* = [1.0] (*abc* == [10]) (*abc* ++){
must be integer type.
7) All variables names must be same as declaration.
run (Print *abc* = 10) (*abe) >= 20) (*def* ++)
X they both must
g) to Newline character boefor a closing '3' is must.
run (
run (){
3 1

10) we are not allowing any Statement within the loop for now.

Yun(______) \{

Point *abc*! \times although of is correct Statement but

we will throw error of the mission

Closing Parathesis.

Variable declaration Rules along with test Cores. 1) Variable name must start and end with * G can contain any
Tracket and alink: (can also start with any).
O * abc.* 1 Varrage .
Variable declaration Variable initiozation
Variable declaration DT [Var name] Variable initiazation Space space Variable initiazation Space space Space
no Point * def *!
4) Once the variable is defined (declared or introlized) can never be redectare again
Mopeint *abc*! X *abc * was declared alredy
5) multiple declaration is not allowed.
Point *abc *, *def*!
6) Accepted datatypes are
Point and mapaint and they are case servitue
Point *abex! X
Poin2t -x abcx! X.
7) A variable can be reffered for initialization after its
Point *abc*! *abc*! \times

8)

	LIBURA AND AND AND AND AND AND AND AND AND AN
A LOCATORY	AITIM SUBAIRAY 209 PEO
1- VARIABLE:	S - C - C - C - C - C - C - C - C - C -
R.E FOR VAR	PIARIF:
$\angle = \{ \times, 0-9, A-Z, \alpha \}$	9
	,-2)
S = *	
L = A - Z	
l = a - z	
d=0-9	$R \cdot E = ^{\times} [a - zA - ZO - 9] \times \times $
R.E = S(L+1+d)+ S	$\sum_{k=1}^{\infty} \frac{1}{k!} \left(\frac{1}{k!} - \frac{1}{k!} - \frac{1}{k!} \right) = \frac{1}{k!} \left(\frac{1}{k!} - \frac{1}{k!} - \frac{1}{k!} \right)$
Example:	41017AY1A3C
* 12a*	
Ab2	LA Da Dalsper toingia
aAf	NAMES A DECRET X SIND A LOND
	13 D. D. x. COV x Jourgest in
2- DATA TYPES:	FARSE RRE:
1) point (0-9)	$R.E = ^{(-+)}[0-9] * (0-9) * $
2) nopoint (0-9) . (0-9	R.F.= ^{-+]?[0-9]*\$
CEC FOO VA	RIABLE DECLARATION:
<pre></pre> <pre>< Var> -> < DT> *</pre>	
<dt> -> nopoint p</dt>	oint
example:	
	1/k regions Desillengaror (1) La 1981 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	B subject to the control of the cont
	ar Kar-fra
	15 1/400 - repairs the depair to trace
$\langle va^2 \rangle \rightarrow \langle DT \rangle \times va$	
The second secon	1011 x 1 x 1 x 2 2 2 2 2 2 2 2 2 2 2 2 2 2
PARSE TREE:	3 1023
	<dys *="" *<="" td="" varname=""></dys>
	Point var1

Dem I [(DVarramo) - Conteg Trovame [cord spec [intega) (Ovarrane CFG FOR VARIABLE INITIALIZATION: Date <init > > <OT> * varrame * <space> = <space> < value>! Kapital your <DT> > point Inopoint (space)) II-> space how < value> > d+ d+ d+ (already var) -> x voirance x au Example: nopoint * var1 * [= [4] point * 20bA * 1 = 1 2.01 DERIVATION: nopoint *var1*1 = 1 4! <init > → < DT> * varname * < space> = < space> < value>! -> nopoint * var1 * [] = [] 4! PARSE TREE: Limit * votrame * (space) = «pace> «Value> cke/dos vasivalist/ thectives of 3- LOOP:-< sun> -> sun espace> (< initializer>) (< condition>) (< inc/de >)] <initializer> -> espace>(var> espace> = espace> d+ espace> ccondition> -> <space> < var> <space> < cond-op> <pace> a d+ sincldes -> sepace> sepace> sepace> sepace> <BYAT> → <DT> × Yarname × \$ <cond-op> -= = | != | > | < | > = | ≤ 33969

DERIVATION: run [(var1 = 4)(var1 < 8)(var1 +4)] <vun>→vun <space>[(<initializer>)(<condition>)(<inc/dec>)] -> run [[(< space > < var > < space > d + < space >) (< space > < var > < space > econd-op> < space > d+) (<space> < vario)++ < space> ?] -> run [[(] * varname *] =] 4]) ([* varname *] < [&] (] * varname* [(0++ -> sun [([+ x 1 x 0 + 0) ([+ var 1 x 0 < 0 8) ([+ x 1 x x 1 x 1 + 1 0)] PARSE TREE:

Keywords:	Le Mannayan
point, nopoint, run, check, otherwise check	c, otherwise
SYMBOLS:	
!,;,(,),{,},{,,},*	
CONDITIONAL:	January Million
NT>-> check == [< condition> < Next>] {< statement	
otherwise_check [<condition> <next< th=""><td>>) { Zstatement >]!</td></next<></condition>	>) { Zstatement >]!
otherwise (statement)	
<pre><condition> > (var) < operator> (val)</condition></pre>	,
<pre><pre><pre><pre></pre></pre></pre></pre>	
 	
$\langle n v_m \rangle \rightarrow d^+$	
<next> -> < logical-op> < condition></next>	1-1-
<log(al.qp> → ll/11)</log(al.qp>	
DEPUIATION OF IT	ξ
DERNATION: Check [*var1 * ==0]	[2 statement 7]
<nt> → Check [< van> < operator> < val>]</nt>	
\rightarrow check [\star varname $\star = = \langle num \rangle$]	
\rightarrow check $[* vay1 * == 0]$	
< NT>	
check ((Var) (Spera	law chall
	1
* varrame * =:	=\ <num></num>
var1	0