```
<start> -> < expr> FOF
 Lexpro> -> 2 term > 2 exprp>
cexprp> -> + 2 term> cexprp>
                                  |-cterm>cexprep | E
<tern> -> < fact> < termp>
ctermp> -> *< fact > < termp> |
                                  /= Ract>=temp> | E
= fact > --> (< expr>) | NUM
 First (< start >) = first (< expre>) = {(, NUM}
 First (< expr>) = First (< term>) = {(INUM}
 First (20xprp) = {+,-}
 First (cterm >) = first (cfact >) = 1 (, NUM)
 First (2 temp>) = {*, /}
 First (= fact >) = 1 (, NUM }
                                   A -> & B
 $ E Follow (2 start >)
                                               <expn> EOF
                               <start > ->
I FOF ? & Follow (=expr>)
 Follow (expres) C follow (expres)
                                <expr> > < tem> < lxprp>
1 +,- ? € Follow (< term >)
                                2 expr) ->
                                               Lterm> <exprp>
                                    11 NULL
Follow (cexprs) = follow (<terms)
                                               <tem><exprp>
                               < exprp> -> +
                                     NULC
 Follow (< exprp>) & Follow (<term>)
                               <exprp> -> +<tem> < exprp>
Fallow (< term>) & Fallow (< termp>)
                                 2 term> -> = fact > = termp>
{ * , / } { Follow (< fact >)
                                 <tern> -> < Pact><termp>
                                 WLC
Follow (eterm>) C Follow (efact>)
                                = termp> -> # < foct > < temp>
                                MULC
 follow (<temps) < Follow (~fact >)
                                < temp> -> *= Rct> < temp>
                               < fact > -> ( < expr>)
 { ) } < Follow (< expr>)
```

<b>×</b>	Fs	ell 200	ر ا ت	υ (	×	)		
< Start >	\$							
cexpr>	E	Of					, )	
<expre>&gt;</expre>	F	of				,	)	
cterm>	Es	⊋F <sub>I</sub>	+,	_		1	)	
ctermp>	Ec	≥F,	+,	_		1	)	
<pre>c fact &gt;</pre>	E	⊃f,	+ ,	- / 7	K,	/,	)	

$A \longrightarrow \alpha$	Guida (A-> 2)
cstart> -> <expr> FQF</expr>	First (2 expr>) = } (, NUM }
<expr> -&gt; cterm&gt; &lt; exprp&gt;</expr>	First (zterm>)= { (, NUM }
cexprp> => + cterm> = exprep>	<b>} + }</b>
	} - }
1-> &	Follow (=exprp>)= {EQF. }}
cterm> -> = fact > = termp>	First (= fact >) = } (, NUM}
ztemp> 1 -> # zfact >= temp>	\ \*\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
	Follow (=termp>) = {FOF,+,-,)}
<pre></pre>	<b>\\ \\ \\</b>
T-> NUM	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

T5 Runzionanti

NUM + (NUM) EOF (NUM) EOF

```
< Prog > = P
< statlist > = 50
                                 NULL (SQP)
< statlistp> = SQp
< stat > = 5
< when list > = Wl
< when eist p> = Wlp
                                 NULL (WRP)

Ulen item > = Wi

< bexpr > = B
< expr > = E
<expr list > = EQ
< expresstp > = Elp
                                 NULL (ERP)
First (x)
  First (P) = First (SQ) = 1 = , print, read, cond, while,
 First (52) = First (5) = | = , print, read, cond, while, |}
 First (SQp)= 1;
 First (S) = 1=, print, read, cond, while, }
 First (W2) = First (Wi) = | when }
 First (WQp) = first (Wi)= when }
 Finst (Wi) = when
 finst (B) = | RELOP}
 First (E) = { +, -, *, /, NUM, 1D }
 Finst (EQ) = Finst (E) = { +, -, *, /, NUM, 10 }
 First (Elp) = First (E) = { +, -, *, /, NUM, 10 }
```

A -> & B B S € Follow (P) P -> [FOF] E Follow (50) se fof 50p 52 ->5 Follow (SD) = Follow (SDP) 52 -> 5 Sep 1; }=finst(Sep) & Follow(s) WAST Follow (SQ) C Follow (S) 52p -> ; 5 52p follow (sep) = Follow(5) 52p -> ; 5 Sep Mile Follow (52p) & Follow (52p) 5 -> = 1D E Follow (5) = Follow (E) 5 -> print(Ee) () } < follow (El) 5 -> Gond We else S {else} E Follow (WI) 5 -> (ond Up 5 else Fallow (5) & Fallow (5) 5 -> while (B) 5 { ) } < Follow (B) 5 -> while (5 (5) ( Fallow (5) 5 -> { 52 } 1 ( + Fo@ow (50) We -> Wi Wep Follow (We) = Follow (Wep) WQ -> (when) = First (wep) E Follow (Wi) NULL Follow (ke) & Follow (ki) Wlp -> wi wep Follow (wep) = Follow (wep) Wep -> Wi Wep {when } = First (wep) C Follow (Wi) NULL Follow (Wep) C Follow (Wi) Wi -> When (B) do 5 1) C FOCOOW (B) Wi -> When 5 Follow (Wi) = Follow (S) B -> Relop F E 1+,-,+,1, NUM, 10 = First (E) < fo@ow (E) B -> Relop F E Fallow (B) = Fallow (E) E -> + ( EQ )
\* ( EQ ) ()) E Follow (EQ) E -> - E E 1+,-1+,/,NUM, 10 = Finst (E) < fo@ow (E) E -> - E E Follow (E) S Follow (E) EQ -> E EQP Fellow (FR) = Follow (ERP) ERP follow (FL) = follow (F) Elp -> E Elp 'NULL Follow (Elp) C Follow (Elp) Follow (Flp) = Follow (F) F&P, NULL

X Follow (X	)
P \$	
5Q	
SQP FOF	
5 EOF, i, alse,	}, when
Wl	
Wlp	
Wi	when
B	
E EOF, , , , , else,	} when +, -, *, /, NUM, ID
EQ )	
Esp )	
$A - > \alpha$	Guida (A> ~)
P -> 5e EQF	First (SQ)=1=, Print, read, cond, while,
52> 5 52p	First (5) = 1=, print, read, comd, while,
50p -> ; 5 50p	
50p-> E	Follow (52p) = { FOF, }}
5 -> = ID E	
5 -> print (EQ)	{print?
5 -> read (1D)	{read}
5 -> comd (Wl) else s	{ Como }
5 -> while (B) 5	{while}
5 -> { 50 }	
we -> Wi wep	Firest (Wi) = { when }
W2p -> Wi W2p	Finst (Wi) = { when }
W2p-> E	Follow (Wlp) = {else}
Wi -> when (B) do S	{ when }
B> Relop (E) (E)	{ relop }
E -> + (EQ)	