The initial test removes as a failure such phrases as:

```
b := (a+b)  if ...
b := A[b]  if ...
if a > then if ...
b := a+b if ...
```

EXP is then called to set E to zero if the preceding delimiter was := as in

```
a := if ...
```

In the case of E being equal to 1, DECTYP is tested. It is normally zero (i.e. we are not in a declaration) and DECSTA is set to /o o. It may however be set to array, as in:

```
real array A [1 : PROC (if ...
```

In this case of course DECSTA must not be changed. A test is then made to see if this is an actual parameter, and if it is E is set to zero.

If E was not equal to 1, a further test is made to remove constructs such as

```
y + if
```

The stack entry saves the state variables ARITH, E and EXPTYP and stacks if to be checked by the following then. ARITH and E are then set up for the Boolean expression in the if clause.

ERRORS

FAIL 67;	if misused
FAIL 100;	if must not be used after log., arith. or rel. operator.
FAIL 68;	if used in declaration other than array declaration.