

Block Management conditional statement has the following syntax.

```
IF (L1) THEN
```

```
<IF-block 1>
```

```
ELSE IF (L2) THEN
```

```
<IF-block 2>
```

```
...
```

```
ELSE
```

```
<ELSE-block>
```

```
END IF
```

where IF, THEN, ELSE - keywords conditional statement block management; L1, L2, - logical expressions; <IF-block 1>, <IF-2 block>, <ELSE-block> - blocks containing one or more operators who are subject.

Jobs conditional operator control block is as follows. Calculate the value of a logical expression of L1. If L1 = .TRUE., It runs <IF-block 1>. Then control is transferred to the END IF.

If L1 = .FALSE., Then <IF-block 1> is performed. Control is passed to the ELSE IF (L2) THEN. Calculate the value of a logical expression L2. If L2 = .TRUE., It runs <IF-block 2>. Then control is transferred to the END IF. So is repeated until it is done one <IF-block>. If no logical expression is not equal .TRUE. and none <IF-block> is not fulfilled, then the <ELSE-block> and control is transferred to the END IF.

Example. Sorting

```
IF (M .GT. 1000) THEN
```

```
WRITE (*, *) 'M> 1000'
```

```
ELSE IF (T .GT. 100) THEN
```

```
WRITE (*, *) '1000> M> 100'
```

```
ELSE IF (T .GT. 10) THEN
```

```
WRITE (*, *) '100> M> 10'
```

```
ELSE IF (T .GT. 0) THEN
```

```
WRITE (*, *) '10> M> 0'
```

```
ELSE
```

```
WRITE (*, *) 'M <0'
```

```
END IF
```

The structure of Conditional operator control block can be changed - simplified if necessary.

Example. Make fragment programs for calculating the expression:

```
IF (X .LE. 3.8) THEN
```

```
Y = 1.7 * X + 3.5
```

```
ELSE
```

```
Y = 5.9 * X ** 2 + 8.3
```

```
END IF
```

The simplest case when you can not use the ELSE - <ELSE-block>, apply only one block IF (L) THEN - <IF-block>.

Example. If $T < 120,6$ ask $ALPHA = 12,3$ and $BETA = 2,2 \cdot 10^{-5}$, and bring these values to the console.

```
IF (T .LT. 120.6) THEN
```

```
ALPHA = 12.3
```

```
BETA = 2.2E-5
```

```
WRITE (*, *) 'ALPHA =', ALPHA, 'BETA =', BETA, 'T =', T
```

```
END IF
```