

Value parameters are treated like local variables and, like these variables, the earlier a parameter is declared the higher address it has in memory. However, unlike variables, the lowest (and the highest) address is fixed and this is fixed at $(IX+2)$ e.g.

```
PROCEDURE test(i : REAL; j : INTEGER);
```

then:

j (allocated first) is at $IX+2$ and $IX+3$.
i is at $IX+4$, $IX+5$, $IX+6$, and $IX+7$.

Variable parameters are treated just like value parameters except that they are always allocated 2 bytes and these 2 bytes contain the address of the variable e.g.

```
PROCEDURE test(i : INTEGER; VAR x : REAL);
```

then:

the reference to x is placed at $IX+2$ and $IX+3$; these locations contain the address where x is stored. The value of i is at $IX+4$ and $IX+5$.

Returned values of functions are placed above the first parameter in memory e.g.

```
FUNCTION test(i : INTEGER) : REAL;
```

then i is at $IX+2$ and $IX+3$ and space is reserved for the returned value at $IX+4$, $IX-5$, $IX-6$ and $IX-7$.