[7.17] Returning more than one result from a function

Functions, by definition, can return only one result. You can get around this limitation by returning the answers in a list, string, matrix or data variable. The calling routine must extract the individual answers from the returned result. For example, to return the three results *a*, *b* and *c*, use

```
return {a,b,c}
```

then extract the individual results with

`{a,b,c}[1]`	to get *a*
`{a,b,c}[2]`	to get *b*
`{a,b,c}[3]`	to get *c*

Note that lists may contain not only numbers and expressions, but other lists and matrices if the elements are expressed as equality expressions, for example

```
return {x=[1,2],y={3,4},z={5,6,w={7,8}}}
```

The assignment variables *w*, *x*, *y* and *z* must not exist in the current folder, or the expressions will be evaluated and the list elements will becomre *true* or *false*. To retrieve the individual results, use the list indices and the *right()* function. For example, if the list above is stored in the variable *result*, then

`right(result[1])`	returns [1,2]
`right(result[2])`	returns {3,4}
`right(result[3])`	returns {5,6,w={7,8}}
`right(right(result[3])[3])`	returns {7,8}

To minimize the chance that the assignment variables exist, you can use international characters. The assignment variables need not be unique, for example, this works:

```
return {ä=[1,2],ä={3,4},ä={5,6,ä={7,8}}}
```

Another method to return more than one result is to use the results as arguments of an *undefined* user function. For example, if your function uses

```
return udf(1Ø,2Ø,{1,2,3})
```

then

```
udf(1Ø,2Ø,{1,2,3})
```

will be returned, as long as *udf()* is not actually defined as a function. You can use the *part()* function to extract the various results. With the expression above,

`part(udf(1Ø,2Ø,{1,2,3}),1)`	returns 10
`part(udf(1Ø,2Ø,{1,2,3}),2)`	returns 20
`part(udf(1Ø,2Ø,{1,2,3}),3)`	returns {1,2,3}

In general, for *part(exp,n)*, the *n*th argument is returned.

(Credit to Glenn E. Fisher and Bhuvanesh Bhatt)