$c := \text{stack}(3,7,15,20)$

$b := \text{stack}(2,3)$

This is just a convenient way to create a column vector which can later easily be edited to let it grow or shrink

$a := \text{stack}(5)$

This creates a 1x1 matrix

Now we make your expressions for x and y functions of three arguments. The formal arguments must not be named a,b and c. As y is dependent on x we have to define it after x and use a function call for x().

$$x(a,b,c) := \sqrt{\frac{5+a-b^5}{1-b\cdot c^3}} \qquad y(a,b,c) := 0.45\cdot x(a,b,c)^{0.3} - 0.4\cdot(c-b)^2$$

We already are able to call those functions with various arguments and compare:

$x(4,3,2) = 3.19 \qquad y(4,3,2) = 0.237$

$x(-12,5,6) = 1.704 \qquad y(-12,5,6) = 0.128$

We might also use one of the vectors defined above as function argument, but we should vectorize the function call (or the function definition itself) iven if its not necessary with the function definiton in this examples:

$$\overrightarrow{y(5,b,7)} = \begin{pmatrix} -9.731 \\ -6.04 \end{pmatrix} \qquad \overrightarrow{y\left[5,b,\begin{pmatrix} 7 \\ 10 \end{pmatrix}\right]} = \begin{pmatrix} -9.731 \\ -19.293 \end{pmatrix}$$

$y(a,b,c) = ▮$ The above approach only works if the provided vectors are of same dimension.

So we can write a small program to loop through all vector elements and display the results

$x(5,1,1) = ▮$ Divide by zero.

Later we will create a table with all values for a variety of args, but if the calculation fails for just one triple of args, the whole table will fail. To avoid this, we can use the "on error" statement.

$$x(a,b,c) := \text{NaN} \;\text{ on error }\; \sqrt{\frac{5+a-b^5}{1-b\cdot c^3}} \qquad y(a,b,c) := \text{"oops"} \;\text{ on error }\; 0.45\cdot x(a,b,c)^{0.3} - 0.4\cdot(c-b)^2$$

Not a must but we can write a combined function to display the arguments and both results

$xy(a,b,c) := \text{augment}(a,b,c,x(a,b,c),y(a,b,c)) \qquad xy(-4,5,6) = (-4 \;\; 5 \;\; 6 \;\; 1.702 \;\; 0.128)$

$xy(5,1,1) = (5 \;\; 1 \;\; 1 \;\; \text{NaN} \;\; \text{NaN})$

Its not possible with your functions but if x() calculates OK and just y() fails we would see the result for x and see "oops" for y. Here x() fails, the result is "Not a Number" and any calculation using NaN yields NaN again and thats what we see for y here.

Now for the table. Arguments a, b, c should be column vectors, but the first three lines will convert a single scalar argument to a 1x1 vector, so the routine may be used with scalars, too.

```
tbl(a,b,c) := | a ← stack(a)  if  rows(a) = 0
              | b ← stack(b)  if  rows(b) = 0
              | c ← stack(c)  if  rows(c) = 0
              | T ← ("a"  "b"  "c"  "x"  "y")
              | for  ia ∈ ORIGIN .. last(a)
              |   for  ib ∈ ORIGIN .. last(b)
              |     for  ic ∈ ORIGIN .. last(c)
              |       T ← stack(T, xy(a_ia, b_ib, c_ic))
              | return  T
```

$$\mathrm{tbl}(a,b,c) = \begin{pmatrix} \text{"a"} & \text{"b"} & \text{"c"} & \text{"x"} & \text{"y"} \\ 5 & 2 & 3 & 0.644 & -5.602 \times 10^{-3} \\ 5 & 2 & 7 & 0.179 & -9.731 \\ 5 & 2 & 15 & 0.057 & -67.409 \\ 5 & 2 & 20 & 0.037 & -129.433 \\ 5 & 3 & 3 & 1.707 & 0.528 \\ 5 & 3 & 7 & 0.476 & -6.04 \\ 5 & 3 & 15 & 0.152 & -57.344 \\ 5 & 3 & 20 & 0.099 & -115.375 \end{pmatrix}$$

Using the vectors defined at the top of the sheet

$$\mathrm{tbl}\left[\begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix}, \begin{pmatrix} -3 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ 5 \\ 12 \end{pmatrix}\right] =$$

Using arbitrary vectors on the fly

"a"	"b"	"c"	"x"	"y"
-2	-3	-1	11.09j	-0.77+0.42j
-2	-3	1	7.84	-5.57
-2	-3	5	0.81	-25.18
-2	-3	12	0.22	-89.72
-2	1	-1	1	-1.15
-2	1	1	NaN	NaN
-2	1	5	0.13j	-6.18+0.11j
-2	1	12	0.03j	-48.25+0.07j
1	-3	-1	11.16j	-0.77+0.42j
1	-3	1	7.89	-5.56
1	-3	5	0.81	-25.18
1	-3	12	0.22	-89.71
1	1	-1	1.58	-1.08
1	1	1	NaN	NaN
1	1	5	0.2j	-6.15+0.13j
1	1	12	0.05j	-48.23+0.09j
3	-3	-1	11.2j	-0.77+0.42j
3	-3	1	7.92	-5.56
3	-3	5	0.82	-25.18
3	-3	12	0.22	-89.71
3	1	-1	1.87	-1.06
3	1	1	NaN	NaN
3	1	5	0.24j	-6.14+0.13j
3	1	12	0.06j	-48.22+0.09j