In [1]:
```
use Array: all except {+};
```

In [2]:
```
use y2025assignment2: all;
```

Assignment2: Rank Polymorphism

Addition

Define your own version of `+`. It should operate on arrays of the same shape only. A version for integer arrays is sufficient! In case the shapes of the arguments do not match, an error should be issued. Note here, that you need to change your `use Array:all;` statement into `use Array: all except {+};`!

In [3]:
```
[1] + [3]
```

```
Dimension:  1
Shape    : <  1>
< 4 >
```

In [4]:
```
[1] + [3,2]
```

```
Dimension:  1
Shape    : <  1>
< 4 >
```

```
*** SAC runtime warning
*** In ./y2025assignment2.sac, line 8, column 1
*** Type pattern error in application of +: the found value of `shp'
in `d:shp' of argument `b' is not equal to the defined value of `sh
p'
```

In [5]:
```
2 + [[1,2],[3,4]]
```

```
*** SAC runtime warning
*** In ./y2025assignment2.sac, line 8, column 1
*** Type pattern error in application of +: the found value of `d' i
n `d:shp' of argument `b' is not equal to the defined value of `d'


*** SAC runtime warning
*** In ./y2025assignment2.sac, line 11, column 37
*** Array `SACp_pinl_3904__flat_68' with shape [ 2, 2] does not adhe
re to `dim == 0' constraint


*** SAC runtime error
*** In ./y2025assignment2.sac, line 11, column 19
*** Primitive function _add_SxS_ guard failed

[SaC kernel] Executable exited with code 1
```

In [6]:

```
21+21
```

```
Dimension:  0
Shape    : < >
 42
```

commonPrefix

Define a function that takes two vectors of integers and returns `true` if and only if one vector is the prefix of the other.

In [7]:

```
commonPrefix ([1], [1,2,3,4])
```

```
Dimension:  0
Shape    : < >
 true
```

In [8]:

```
commonPrefix (3,[3,4])
```

```
//var/folders/2t/gqvy03mn5jlgms6b0jffctt00000gp/T/jup-sacfhywzw_s/tm
pxs7mesa4.sac:15: error:
  All instances of "main" contain type errors
No matching definition found for the application of "y2025assignment
2::commonPrefix" to arguments ( int{3}, int[2]{3,4})
Compilation failed while Running type inference system, 1 error(s).
[SaC kernel] sac2c exited with code 52, the executable will not be e
xecuted
```

In [9]:

```
commonPrefix ([2,4], [])
```

```
Dimension:  0
Shape    : < >
 true
```

In [10]:

```
commonPrefix ([2,4],[2])
```

```
Dimension:  0
Shape    : < >
 true
```

commonPostfix

Define a function that takes two vectors of integers and returns true if and only if one vector is the postfix of the other.

```
In [11]: commonPostfix ([2,4,8], [4,8])
```

```
Dimension:  0
Shape    : < >
 true
```

```
In [12]: commonPostfix ([2,4,7], [2,4])
```

```
Dimension:  0
Shape    : < >
 false
```

^+

Now define an addition that is applicable to two arrays of arbitrary shape, provided that the shape of one is a prefix of the shape of the other. If so, replicate the lower-dimensional array on the `inside` so that you can add the two arrays.

```
In [14]: [2,1] ^+ [[3,7,1], [1,2,3]]
```

```
Dimension:  2
Shape    : <  2,  3>
| 5  9  3 |
| 2  3  4 |
```

```
In [15]: [[3,7,1], [1,2,3]] ^+ [2,1]
```

```
Dimension:  2
Shape    : <  2,  3>
| 5  9  3 |
| 2  3  4 |
```

```
In [17]: [3,2,1] ^+ [[3,7,1], [1,2,3]]
```

```
Dimension:  2
Shape    : <  2,  3>
| 6 10  4 |
| 3  4  5 |
*** SAC runtime warning
*** In ./y2025assignment2.sac, line 27, column 1
*** Type pattern pre-condition of ^+ failed


*** SAC runtime warning
*** In ./y2025assignment2.sac, line 8, column 1
*** Type pattern error in application of +: the found value of `shp'
in `d:shp' of argument `b' is not equal to the defined value of `sh
p'


*** SAC runtime warning
*** In ./y2025assignment2.sac, line 8, column 1
*** Type pattern error in definition of +: the found value of `shp'
in `d:shp' of return value is not equal to the defined value of `sh
p'
```

In [45]:

```
1 ^+ [[3,7,1], [1,2,3]]
```

```
Dimension:  2
Shape    : <  2,  3>
| 4  8  2 |
| 2  3  4 |
```

+^

Finally, define an addition that is applicable to two arrays of arbitrary shape, provided that the shape of one is a postfix of the shape of the other. If so, replicate the lower-dimensional array on the `outside` so that you can add the two arrays.

In [47]:

```
[3,2,1] +^ [[3,7,1], [1,2,3]]
```

```
Dimension:  2
Shape    : <  2,  3>
| 6  9  2 |
| 4  4  4 |
```

In [48]:

```
1 +^ [[3,7,1], [1,2,3]]
```

```
Dimension:  2
Shape    : <  2,  3>
| 4  8  2 |
| 2  3  4 |
```

In [ ]: