

Advanced Programming

Martijn Vogelaar 1047391
Gianni Monteban 1047546

February 5, 2021

Exercise 5. Assuming `mat` to be defined as in the previous example, what results do you expect from the following expressions:

```
reshape([3,0,5], [])[[[]]]
```

```
dim(3)  
shape<3,0,5>  
<>
```

```
reshape([3,0,5], [])[[1]]
```

```
dim(2)  
shape(0,5)  
<>
```

```
reshape([3,0,5], [])[[1,0]]
```

error, acces at non existing part

```
mat[reshape([2,0], [])]
```

can't be printed ??

Exercise 6. What results do you expect from the following expressions:

```
min(reshape([3,0,5], []), 42)
```

```
dim(3)  
shape([3,0,5])  
<>
```

```
reshape([3,0,5], []) + reshape([3,0,5], [])
```

```
dim(3)  
shape([3,0,5])  
<>
```

```
reshape([1,1], [1]) + reshape([1], [1])
```

error since the shapes are different

Exercise 7. Which of the following expressions can be reformulated in terms of `take`, `++`, and the basic operations defined in the previous parts?

drop (v, a)

It is possible with the following formula:

```
take((abs(v)/-v) * ((shape(vect) - (v*v/(abs(v))))),vect)
```

tile (v, o, a)

impossible, due to the offset it is impossible to just take the middle part of an array.

shift ([n], e, a)

```
arr = [n]
take(shape(vect),take(-shape(vect)-arr,vect))
```

shift ([m,n], e, a)

```
arr = [m,n]
take(shape(mat),take(-shape(mat)-arr,mat))
```

rotate ([n], a)

We couldn't rewrite it to a `take` or `++`. But we could write it into a `drop`

```
drop(shape(vect) -n,vect) ++ drop(-n,vect) ++ drop(-shape(vect) -n,vect)
```

rotate ([m,n], a)

It should be possible only we couldn't find it.

Can we define the general versions of shift and rotate as well?

shift:

```
take(shape(vect),take(-shape(vect)-v,vect))
```

Exercise 8. All operations introduced in this part apply to all elements of the array they are applied to. Given the array operations introduced so far, can you specify row-wise or column-wise summations for matrices? Try to specify these operations for a 2 by 3 matrix first.

```
mat = [1,2,3,4,5,6];
mat = reshape([2,3], mat);

print(sum(tile([2,1],[0,0],mat)));
print(sum(tile([2,1],[0,1],mat)));
print(sum(tile([2,1],[0,2],mat)));

print(sum(tile([1,3],[0,0],mat)));
print(sum(tile([1,3],[1,0],mat)));
```