Operating on Lists

Created using Maple 14.01

Jake Bobowski

```
> restart;
  with(StringTools) :
  FormatTime("%m−%d−%Y, %H:%M");
```

"10-11-2013, 13:32" **(1)**

This Maple input enters a list of decimal numbers

```
> listA := [100, 100, 97, 93, 75.5, 46.8, 27.5, 19.1, 13.4, 9.9];
```

listA := [100, 100, 97, 93, 75.5, 46.8, 27.5, 19.1, 13.4, 9.9] **(2)**

The number of points in the list can be found using the *nops*( ) function

```
> nops(listA);
```

10 **(3)**

Any element of the list can be accessed using the following Maple input

```
> listA[1];
  listA[2];
  listA[5];
  listA[nops(listA)];
```

100
100
75.5
9.9 **(4)**

Certain operations on a list do what you might expect

```
> 10· listA;
```

[1000, 1000, 970, 930, 755.0, 468.0, 275.0, 191.0, 134.0, 99.0] **(5)**

However, others may not

```
> listA·listA;
  sqrt(listA);
  sin(listA);
```

$[100, 100, 97, 93, 75.5, 46.8, 27.5, 19.1, 13.4, 9.9]^2$

```
Error, invalid input: sqrt expects its 1st argument, x, to be of
type algebraic, but received [100, 100, 97, 93, 75.5, 46.8,
27.5, 19.1, 13.4, 9.9]
Error, invalid input: sin expects its 1st argument, x, to be of
type algebraic, but received [100, 100, 97, 93, 75.5, 46.8,
27.5, 19.1, 13.4, 9.9]
```

The *seq* function is extremely useful for completing these operations. The *i*=1..5 notation increments *i* from 1 to 5 in steps of 1. *i*=1..10,2 will incremet *i* from 1 to 10 in steps of 2.

```
> seq(i, i = 1 ..5);
  seq(i, i = 1 ..10, 2);
  seq(listA[i], i = 1 ..nops(listA));
```

1, 2, 3, 4, 5

1, 3, 5, 7, 9

100, 100, 97, 93, 75.5, 46.8, 27.5, 19.1, 13.4, 9.9 **(6)**

Note that *seq* can be used to create a list by putting the whole thing inside square brackets [*seq*(...)]

```
> listB := [seq(i, i = 1..5)];
  listB[1];
  listB[nops(listB)];
```

$$listB := [1, 2, 3, 4, 5]$$

1

5 **(7)**

Now let's use *seq* to complete the three operation and failed above. Note that the *evalf* command is used to force Maple to display a numerical value rather than $\sqrt{97}$ (for example).

```
> listAsquared := [seq(listA[i]^2, i = 1..nops(listA))];
  listAsqrt := [seq(evalf(sqrt(listA[i]))), i = 1..nops(listA))];
  listAsin := [seq(evalf(sin(listA[i]))), i = 1..nops(listA))]
```

$$listAsquared := [10000, 10000, 9409, 8649, 5700.25, 2190.24, 756.25, 364.81, 179.56, 98.01]$$

$$listAsqrt := [10., 10., 9.848857802, 9.643650761, 8.689073598, 6.841052551, 5.244044241, 4.370354677, 3.660601044, 3.146426545]$$

$$listAsin := [-0.5063656411, -0.5063656411, 0.3796077390, -0.9482821413, 0.1016006979, 0.3182565111, 0.6992400317, 0.2478342080, 0.7403758900, -0.4575358938] \quad \textbf{(8)}$$

In this final example will demonstrate using nested *seq* commands. Let's suppose that you wanted to make a list like [0,0,0,0,1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4, ...] up to 100 without have to manually type everything and with out having to manually combine 101 individual lists. We will use on *seq* statement inside another *seq* statement to accomplish this task. The inner *seq* is used to create the short sequences of 0,0,0,0 and 1,1,1,1 and so on, while the outer *seq* is used to increment the value of the number used by the inner *seq*.

```
> nestedList := [seq(seq(j, i = 1..4), j = 0..100)];
```

$$nestedList := [0, 0, 0, 0, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 6, 6, 7, 7, 7, 7, \quad \textbf{(9)}$$
$$8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 11, 11, 11, 11, 12, 12, 12, 12, 13, 13, 13, 13, 14, 14, 14,$$
$$14, 15, 15, 15, 15, 16, 16, 16, 16, 17, 17, 17, 17, 18, 18, 18, 18, 19, 19, 19, 19, 20, 20, 20,$$
$$20, 21, 21, 21, 21, 22, 22, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25, 25, 25, 25, 26, 26, 26,$$
$$26, 27, 27, 27, 27, 28, 28, 28, 28, 29, 29, 29, 29, 30, 30, 30, 30, 31, 31, 31, 31, 32, 32, 32,$$
$$32, 33, 33, 33, 33, 34, 34, 34, 34, 35, 35, 35, 35, 36, 36, 36, 36, 37, 37, 37, 37, 38, 38, 38,$$
$$38, 39, 39, 39, 39, 40, 40, 40, 40, 41, 41, 41, 41, 42, 42, 42, 42, 43, 43, 43, 43, 44, 44, 44,$$
$$44, 45, 45, 45, 45, 46, 46, 46, 46, 47, 47, 47, 47, 48, 48, 48, 48, 49, 49, 49, 49, 50, 50, 50,$$
$$50, 51, 51, 51, 51, 52, 52, 52, 52, 53, 53, 53, 53, 54, 54, 54, 54, 55, 55, 55, 55, 56, 56, 56,$$
$$56, 57, 57, 57, 57, 58, 58, 58, 58, 59, 59, 59, 59, 60, 60, 60, 60, 61, 61, 61, 61, 62, 62, 62,$$
$$62, 63, 63, 63, 63, 64, 64, 64, 64, 65, 65, 65, 65, 66, 66, 66, 66, 67, 67, 67, 67, 68, 68, 68,$$
$$68, 69, 69, 69, 69, 70, 70, 70, 70, 71, 71, 71, 71, 72, 72, 72, 72, 73, 73, 73, 73, 74, 74, 74,$$
$$74, 75, 75, 75, 75, 76, 76, 76, 76, 77, 77, 77, 77, 78, 78, 78, 78, 79, 79, 79, 79, 80, 80, 80,$$
$$80, 81, 81, 81, 81, 82, 82, 82, 82, 83, 83, 83, 83, 84, 84, 84, 84, 85, 85, 85, 85, 86, 86, 86,$$
$$86, 87, 87, 87, 87, 88, 88, 88, 88, 89, 89, 89, 89, 90, 90, 90, 90, 91, 91, 91, 91, 92, 92, 92,$$

92, 93, 93, 93, 93, 94, 94, 94, 94, 95, 95, 95, 95, 96, 96, 96, 96, 97, 97, 97, 97, 98, 98, 98, 98, 99, 99, 99, 99, 100, 100, 100, 100]

>