▶ Initial setup & conversion of text to/from various list formats.

```
> n:=15;
                              n := 15                                   (1)
> seq(2^k mod n, k = 1 ..20);
           2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, 8, 1      (2)
> seq(3^k mod n, k = 1 ..20);
        3, 9, 12, 6, 3, 9, 12, 6, 3, 9, 12, 6, 3, 9, 12, 6, 3, 9, 12, 6   (3)
> seq(4^k mod n, k = 1 ..20);
           4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1, 4, 1      (4)
> seq(5^k mod n, k = 1 ..20);
     5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10, 5, 10 (5)
> seq(6^k mod n, k = 1 ..20);
           6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6      (6)
> seq(7^k mod n, k = 1 ..20);
        7, 4, 13, 1, 7, 4, 13, 1, 7, 4, 13, 1, 7, 4, 13, 1, 7, 4, 13, 1   (7)
> seq(8^k mod n, k = 1 ..20);
           8, 4, 2, 1, 8, 4, 2, 1, 8, 4, 2, 1, 8, 4, 2, 1, 8, 4, 2, 1      (8)
> seq(9^k mod n, k = 1 ..20);
           9, 6, 9, 6, 9, 6, 9, 6, 9, 6, 9, 6, 9, 6, 9, 6, 9, 6, 9, 6      (9)
> seq(10^k mod n, k = 1 ..20);
  10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10   (10)
> seq(11^k mod n, k = 1 ..20);
     11, 1, 11, 1, 11, 1, 11, 1, 11, 1, 11, 1, 11, 1, 11, 1, 11, 1, 11, 1 (11)
> seq(12^k mod n, k = 1 ..20);
        12, 9, 3, 6, 12, 9, 3, 6, 12, 9, 3, 6, 12, 9, 3, 6, 12, 9, 3, 6   (12)
> seq(13^k mod n, k = 1 ..20);
        13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1, 13, 4, 7, 1   (13)
> seq(14^k mod n, k = 1 ..20);
     14, 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, 1, 14, 1 (14)
> n := 77;
                              n := 77                                   (15)
> seq(2^k mod n, k = 1 ..40);
2, 4, 8, 16, 32, 64, 51, 25, 50, 23, 46, 15, 30, 60, 43, 9, 18, 36, 72, 67, 57, 37, 74, 71, 65, 53, 29,   (16)
    58, 39, 1, 2, 4, 8, 16, 32, 64, 51, 25, 50, 23
> seq(3^k mod n, k = 1 ..40);
3, 9, 27, 4, 12, 36, 31, 16, 48, 67, 47, 64, 38, 37, 34, 25, 75, 71, 59, 23, 69, 53, 5, 15, 45, 58, 20,   (17)
    60, 26, 1, 3, 9, 27, 4, 12, 36, 31, 16, 48, 67
> seq(4^k mod n, k = 1 ..40);
4, 16, 64, 25, 23, 15, 60, 9, 36, 67, 37, 71, 53, 58, 1, 4, 16, 64, 25, 23, 15, 60, 9, 36, 67, 37, 71,   (18)
```

```
    53, 58, 1, 4, 16, 64, 25, 23, 15, 60, 9, 36, 67
>
>
>
> (2^3)^5 mod 15;
                                   8                                   (19)
> 2^8 mod 15
                                   1                                   (20)
> 2^(11 + 7) mod 77;
                                  36                                   (21)
> 5^2 mod 6;
                                   1                                   (22)
>
>
>
> p := nextprime(10);
                                p := 11                                (23)
> q := nextprime(11);
                                q := 13                                (24)
> n := p·q;
                               n := 143                                (25)
> phi := (p − 1 )·(q − 1);
                               ϕ := 120                                (26)
> e := 47;
                                e := 47                                (27)
> d := 1/e mod phi;
                                d := 23                                (28)
> meow := StringToList("cat");
                          meow := [67, 65, 84]                         (29)
> map( x→ x^e mod n, meow);
                             [111, 65, 50]                             (30)
> map(y→y^d mod n, %);
                              [67, 65, 84]                             (31)
> ListToString(%);
                                 "cat"                                 (32)
>
>
>
> big := rand(10^100..10^101) :
> p := nextprime(big( ));
p :=                                                                   (33)
```

```
    76426880842275697096283368069929924187220926409565667330296641985216015045\
    247402583187728969037104233
> q := nextprime(big( ));
q :=                                                                        (34)
    85334579437521204837580966498416610086705384253243464752166801900511414455\
    707593294960500604536421913
> n := p·q;
n :=                                                                        (35)
    65218557343971430017335614947873146442853497017401161523293020441876978915\
    64093504601276869303710623570286950311307222784713565202861152098267427472\
    683707657351089417069443779552952627903857952646257729
> ifactor(n);
Warning,  computation interrupted
> phi := (p − 1)·(q − 1);
ϕ :=                                                                        (36)
    65218557343971430017335614947873146442853497017401161523293020441876978915\
    64093504601276869303710623408525490031510320850849230634514617824341116809\
    874575574887645531342014278597956749755628379072731584
> e := 47;
                              e := 47                                       (37)
> d := 1/e mod phi;
d :=                                                                        (38)
    34690721991474164902838093057379333214283775009255936980475010873338818572\
    14943353511317483672186501813045473421016128112153846082188626502309104686\
    103497646216832729437241637552104654125334244187623183
> meow := StringToList("cat");
                          meow := [67, 65, 84]                              (39)
> map( x→ x^e mod n, meow);
[                                                                           (40)
    66913805084570123447558018091431659442943981023312607454084221003235794522\
    457873706923,
    16103777528925348059351720308521686828629366232055784990961910807527601718\
    902587890625,
    27613817159246019581537952665001988827873527731919498046546135997194165822\
    44579654598918144]
> map(x->x&^d mod n, %);
                            [67, 65, 84]                                    (41)
>
Error, (in unknown) numeric exception: overflow
> ListToString(%);
>
```