The following is a demonstration of RSA public key encryption. Alice makes the following preparations so that others will be able to send her secure messages over an open communication channel. She chooses two large primes p and q at random. She sets $n = pq$ and computes the value of $\phi(n) = (p-1)(q-1)$.

```
> p:=nextprime(rand(10^100)());
> q:=nextprime(rand(10^100)());
> n:=p*q;
> phi_n:=(p-1)*(q-1);
```

p := 400340159470592317831490619591482513697328131486228945410074523776903441005\
7080703111299605127114623

q := 992889151524251562032482805591285422750752571798135144747357026298149152779\
7413449568788992987500477

n := 397493401257839969865622169609477903896759721352381988637644413181737938837\
963474799230874458185109108453565855929133656205035384971875413029348806346987\
463102579267751580337049180294375673518461751 71

phi_n := 397493401257839969865622169609477903896759721352381988637644413181737938\
8837963474799230874458185108969130634756444745669807692866695081768221278477 91\
057844735942426024632095850423876757478753731560072

Alice chooses a random number e less than n. She checks that $\gcd(e, \phi(n)) = 1$ (if this fails, she finds a new value of e that works).

```
> e:=rand(n)();
> gcd(e,phi_n);
```

e := 213774991489513250791488127322381708473736109000630946485643309547324013983\
872496583670255116870544830904977563683579962892793633611384723484293876675165\
736320304346866990155380240989119259842262351 60

8

```
> e:=rand(n)();
> gcd(e,phi_n);
```

e := 349067386640547027881723105408346914113340387424664698822145917141481101818\
280165893561455411359835312925598821203638016231077871111812932102387148353571\
685800994818283355617848132553372510073948199 25

1

Alice sets d equal to the inverse of e mod p (which her computer finds using Eulcid's algorithm).

```
> d:=1/e mod phi_n;
```

d := 988615890152025031325078917963659872755898492630564244953837065057570535240\
212714349787420770434190330938993785823539017799213387672063599042935546343444\
892915852765274652548466093514874941153883035 7

Alice publishes the values of n and e (her public encryption key). She keeps the other values (p, q, $\phi(n)$ and d) secret; the value of d is her decryption key.

Bob wants to send Alice an encrypted message over an insecure channel, which only she can decrypt. His message, "SEND MONEY", is first translated into an integer m using A=01, B=02, ..., Z=26, blank=27, giving

```
> m:=19051404271315140525;
```

$$m := 19051404271315140525$$

(If m is larger than n, he must split up this string of digits into pieces, each of which is smaller than n, and encrypt each piece of the message separately.) Bob looks up Alice's public key (n, e) posted on her website. He encrypts his message m by raising it to the power e mod n, which his computer evaluates using fast modular exponentiation.

```
> m1:=m&^e mod n;
```

$$m1 := 28776989480917770993103038373399274316024986185779418363559509248846196592228710586576786131193428586576319219703258122684895688576804008232223119785546188986719974299364788929279495753989373528618399$$

Bob sends the resulting encrypted message m_1 to Alice over the insecure communication channel.

Alice receives the encrypted message m_1. To decrypt it, she raises m_1 to the power d mod n, also using modular exponentiation:

```
> m2:=m1&^d mod n;
```

$$m2 := 19051404271315140525$$

The resulting integer m_2 which she obtains, coincides with Bob's original message m.