More Discrete Log Calculations

```
r:=random(10^9): p:=nextprime(r())
```

633073699

```
alpha:=numlib::primroot(p)
```

3

Let's say we want the discrete log of 123456789. Trial and error search could in principle work, but it's pretty slow:

```
for j from 1 to p-1 do
if powermod(alpha, j, p) = 123456789 then print(j); break end_if
end_for
Error: Computation aborted
```

I gave up after 100 seconds of CPU time. Perhaps a better approach is Pohlig-Hellman, to the extent that it's feasible:

```
ifactor(p-1)
```

$$2\,3^{3}\,421\,27847$$

Mod 2 is easy:

```
numlib::legendre(123456789,p)
```

-1

So the log is 1 mod 2.

```
c:=powermod(123456789, (p-1)/9, p);
for j from 0 to 8 do
  t:=powermod(alpha, j*(p-1)/9, p);
  if t=c then print(j, t); break end_if;
end_for
```

82102146

7, 82102146

The log is 7 mod 9.

```
c:=powermod(123456789, (p-1)/421, p);
for j from 0 to 420 do
  t:=powermod(alpha, j*(p-1)/421, p);
  if t=c then print(j, t); break end_if;
end_for
```

453820832

202, 453820832

The log is 202 mod 421.

The log is 202 mod 421.

```
c:=powermod(123456789, (p-1)/27847, p);
for j from 0 to 27846 do
  t:=powermod(alpha, j*(p-1)/27847, p);
  if t=c then print(j, t); break end_if;
end_for
```

```
  579256292
```

```
  26605, 579256292
```

The log is 26605 mod 27847. Finally,

```
L := numlib::ichrem([1, 7, 202, 26605], [2, 9, 421, 27847])
```

```
  50819533
```

Check:

```
powermod(alpha, L, p)
```

```
  123456789
```

Another Example

Now let's try again with a larger prime for which p-1 doesn't have such small factors. Even so, the method works.

```
r:=random(10^10): p:=nextprime(r())
```

```
  2062222087
```

```
ifactor(p-1)
```

```
  2 3 827 415603
```

```
alpha:=numlib::primroot(p)
```

```
  3
```

One of the prime factors of p-1 is now 6 digits instead of 5.

```
c:=powermod(123456789, (p-1)/415603, p);
for j from 0 to 415602 do
  t:=powermod(alpha, j*(p-1)/415603, p);
  if t=c then print(j, t); break end_if;
end_for
```

```
  518842466
```

```
  307627, 518842466
```

This took 14 seconds of CPU time on my machine. Finding L mod the other factors is easy.

```
numlib::legendre(123456789,p)
```

```
-1
```

```
c:=powermod(123456789, (p-1)/3, p);
for j from 0 to 2 do
  t:=powermod(alpha, j*(p-1)/3, p);
  if t=c then print(j, t); break end_if;
end_for
```

```
944642688


1, 944642688
```

```
c:=powermod(123456789, (p-1)/827, p);
for j from 0 to 826 do
  t:=powermod(alpha, j*(p-1)/827, p);
  if t=c then print(j, t); break end_if;
end_for
```

```
2006178524


58, 2006178524
```

Finally,

```
L := numlib::ichrem([1, 1, 58, 307627], [2, 3, 827, 415603])
```

```
30231043
```

Check:

```
powermod(alpha, L, p)
```

```
123456789
```