Plotting with GP

You can also make simple plots with GP, such as

```
? ploth(t=0,Pi*2,[sin(t*17)*13,cos(t*52)],1)
%18 = [-12.999928624394538 4, 12.999928624394538 4, -0.99999780382812719 6
?
```

The final "1" indicates that this is plotted as a two-dimensional parametric function, i.e., the x coordinate is $x = sin(17t)$, and the y coordinate is $y = cos(52t)$.

Programming with GP

You can program inside of the gp shell. The basic control structures are

```
while(CONDITION,CODE)

if(CONDITION,THEN-CODE,ELSE-CODE)

for(VAR=A,B,CODE)

forstep(VAR=A,B,STEP,CODE)
```

Examples

```
? for(i=2,10,print(i*i%(i+10)))
4
9
2
10
4
15
10
5
0
```

Examples

```
? forstep(i=1,6,0.5,print(i))
1
1.5000000000000000000000000000
2.0000000000000000000000000000
2.5000000000000000000000000000
3.0000000000000000000000000000
3.5000000000000000000000000000
4.0000000000000000000000000000
4.5000000000000000000000000000
5.0000000000000000000000000000
5.5000000000000000000000000000
6.0000000000000000000000000000
?
```

Examples

```
? x = 0
%1 = 0
? while(x < 10, x = x+1; print(x))
1
2
3
4
5
6
7
8
9
10
?
```

Examples

```
? x = 10
  %1 = 10
? while(x > 0, if(x % 2 == 0, x = x / 2 , x = x + 7); print(x))
5
12
6
3
10
5
12
  [ ... ]
```

Defining functions

Function definition syntax:

```
NAME([ARG1, [ARG2, [...]]]) = local([ARG1, [ARG2, [...]]]) ; CODE


NAME([ARG1, [ARG2, [...]]]) =
{
  local([ARG1, [ARG2, [...]]]) ; CODE
}
```

Examples

```
/* long form */
? first_prime_div(x) =
{
  forprime(p=2,x,if(x % p == 0, return(p)))
}
? first_prime_div(35)
%19 = 5
?

/* short form */
? first_prime_div2(x) = forprime(p=2,x,if(x % p == 0, return(p)))
? first_prime_div2(161)
%20 = 7
```