

"fp.pl October 16, 1978 10:53 AM";

```

Apply ← ([x, f]: x/f);
Plus ← (x: x/plus); "to get around Poplar fixup feature";
"Sub1 ← (x: x-1)";
Times ← (x: x/times); "to get around Poplar fixup feature";
First ← (x: 1/x); x/1
Second ← (x: 2/x); x/2
Rest ← (x: -1/x); x/-1
"Appendl ← ([l, r]: [l], r)";
"Appendr ← ([l, r]: l, [r])";
"Distl ← ([l, r]: r/(x: [l, x]))";
"Distr ← ([l, r]: 1/(x: [x, r]))";
K ← (f: g: f); ident
I ← (x: x); "to get around Poplar fixup feature";
B ← (f: g: x: x/g/f);
Y ← (f: (g: x: x/(g/g)/f) / (g: x: x/(g/g)/f));
CN ← ([p, f, g]: x: ((x/p)>(x/f)) | (x/g)); x/p > x/f | x/g
"WHILE ← (p: f: x: x % {x: (x/p)>(x/f)})";
COMP ← (f: (f/isnull)>I | (f/First/(f/Rest/COMP/B)));
"AA ← (f: x: x/f)";
NF ← (f: x: f/(f: x/f)); x/f
"INS ← (f: x: x///f)";
MC ← (f: x: [x, f/MC]/f);
"Tr ← (x: (1/x/isnull) > [] | ([x//First], (x//Rest/Tr)))";
"Ip ← [Tr, Times/AA, Plus/Ins]/COMP";
"Mm ← [[First, Second/(Tr/B)]/NF, Distr, Distl/AA, Ip/Aa/Aa]/COMP";
Eq ← ([l, r]: (1/isstring)>(r/isstring)>(1/{r}) | (mx)
  ((1/isnull)>(r/isnull)) |
  ((1/isfail)>(r/isfail)));
Equal ← ([l, r]: ([l, r]/Eq) |
  ((1/islist)>(r/islist)>([l/length, r/length]/Eq)>~([l, r]/marry//Equal/isnull)));
Fact ← [
  [[First, 0/K]/NF, Eq]/COMP,
  1/K,
  [[First, [[[[First, -1/K]/NF, Plus]/COMP, Second]/NF, Apply]/COMP]/NF, Times]/COMP
  ]/CN/MC;

```

Eq ← ([l, r]: ~l > ~r | l/isstring > r/isstring > l/{r} | l/islist > r/islist > [l, r]/EqList)

EqList ←

Poplar Version 0.0 Welcomes You

\$\$\$p

[\$Fact, 0/Fact, 1/Fact]/write "t\$"

Error: Input to write routine must be string, in:

[\$Fact, "0"/Fact, "1"/Fact]/write "t\$"

Bigger context? No.

[\$Fact, 0/Fact, 1/Fact]/listout "t\$"

\$1--10///times

\$1--11///times

\$1--12///times

\$1--13///times

\$quit

```

[
(x:[
x,
(f:(g:(x:x/[
(x:x/(x:[
(x:"1"/x),
(g:"0")
]//(f:x/f))/(x:x/(
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/(
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/f|x/g)))
]/(f:(g:(x:x/[
(x:x/(x:[
(x:"1"/x),
(g:"0")
]//(f:x/f))/(x:x/(
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/(
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/f|x/g))),
(g:(x:x/[
(x:x/(x:[
(x:"1"/x),
(g:"0")
]//(f:x/f))/(x:x/(
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/(
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/[
"0",
(f:(g:(x:x/[
(x:x/(x:[
(x:"1"/x),

```

```

(g:"0")
]//(f:x/f))/(x:x/([
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/([
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/f|x/g)))
]|x/g)),
(g:(x:x/[
(x:x/(x:[
(x:"1"/x),
(g:"0")
]//(f:x/f))/(x:x/([
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/([
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/[
"1",
(f:(g:(x:x/[
(x:x/(x:[
(x:"1"/x),
(g:"0")
]//(f:x/f))/(x:x/([
l,
r
]:l/isstring>r/isstring>l/{r}|l/isnull>r/isnull|l/isfail>r/isfail)/(x:x))),
(g:"1"),
(x:x/(x:[
(x:"1"/x),
(x:x/(x:[
(x:x/(x:[
(x:"1"/x),
(g:"-1")
]//(f:x/f))/(x:x/(x:x/times)/(x:x))),
(x:"2"/x)
]//(f:x/f))/(x:x/([
x,
f
]:x/f)/(x:x)))
]//(f:x/f))/(x:x/(x:x/times)/(x:x)))
]>x/f|x/g)))
]|x/g))
]

```