

$a^x - b^y = c$ bootstrapping sieve a, b composite

15 Dec 2007

22 Dec 2007

24 Dec 2007

29 Dec 2007

We are testing for $a^{x_2} - a^{x_1} = b^{y_2} - b^{y_1}$ for $(a, b) = 1$
with the conditions

$b < 100$,

$a < 2411.31 \log(b)$

Rewrite it as $a^{x_1} (a^{(x_2-x_1)} - 1) = b^{y_1} (b^{(y_2-y_1)} - 1)$

We assume that $x_1 > m$ so we begin at $x_1 = 2$, hence we can assume that a divides $b^{y_2-y_1}$.

If $a^{x_2-x_1} > 3798 \log(b)$ then we have exceeded our bound and so know there is no solution.

```
> restart; with(numtheory) :
> ithprime(50); ithprime(1000); ithprime(100000);
      229
      7919
     1299709
(1)

> lowestfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i from 1 to 50 do
    if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
  od;
  prlist;
end;
lowestfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i to 50 do
    if mod(n, ithprime(i)) = 0 then prlist := { ithprime(i), op(prlist) } end if;
  end do;
  prlist;
end proc
(2)

> lowerfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i from 1 to 1000 do
    if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
  od;
```

```

prlist;
end;
lowerfactors := proc(n)
local i, prlist;
prlist := { };
for i to 1000 do
if mod(n, ithprime(i)) = 0 then prlist := {ithprime(i), op(prlist)} end if
end do;
prlist
end proc

```

(3)

```

> lowfactors := proc(n)
local i, prlist;
prlist := { };
for i from 1 to 10000 do
if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
od;
prlist;
end;

```

```

lowfactors := proc(n)
local i, prlist;
prlist := { };
for i to 10000 do
if mod(n, ithprime(i)) = 0 then prlist := {ithprime(i), op(prlist)} end if
end do;
prlist
end proc

```

(4)

```

>
> morefactors := proc (n)
local i, prlist;
prlist := { };
for i to 100000 do
if mod(n, ithprime(i)) = 0 then
prlist := {ithprime(i), op(prlist)}
end if
end do;
prlist
end proc;

```

```

morefactors := proc(n)
local i, prlist;
prlist := { };
for i to 100000 do
if mod(n, ithprime(i)) = 0 then prlist := {ithprime(i), op(prlist)} end if
end do;
prlist

```

(5)

end proc

>

```
> bootstrap4fold := proc(a,b)
  local prlista1, prlista2, prlista3, prlista4, prlistb1,
  prlistb2, prlistb3, prlistb4, i, j, p, aexp, bexp, x2bound,
  fini;
  x2bound:= evalf(3798*log(b));
  fini:= false;
  #first step; this is just a repeat of checkaexp and checkbexp.
  aexp := order(a,b); bexp:= ilcm(a, order(b,a)); #note we
  assume a divides bexp
  prlista1:= lowestfactors(a&^aexp - 1) minus factorset(2*b);
  if nops(prlista1)> 0 then
    for i from 1 to nops(prlista1) do
      bexp := lcm(bexp, order(b, prlista1[i]));
    od;
  fi;
  #we assume a divides the exponent of b
  prlistb1:= lowestfactors(b&^bexp - 1) minus factorset(2*a);
  if nops(prlistb1)> 0 then
    for i from 1 to nops(prlistb1) do
      aexp := lcm(aexp, order(a, prlistb1[i]));
      if aexp > x2bound then fini := true; end if;
    od;
  fi;

  if fini = false then
    #step 2
    prlista2:= lowerfactors(a &^ aexp - 1) minus (prlista1 union
    factorset(2*b));
    if nops(prlista2)> 0 then
      for i from 1 to nops(prlista2) do
        bexp := lcm(bexp, order(b, prlista2[i]));
      od;
    fi;
    prlistb2:= lowerfactors(b &^ bexp - 1) minus (prlistb1 union
    factorset(2*a));
    if nops(prlistb2)> 0 then
      for i from 1 to nops(prlistb2) while fini = false do
        aexp := lcm(aexp, order(a, prlistb2[i]));
        if aexp > x2bound then fini := true; end if;
      od;
    fi;
  fi;
end proc;
```

```

fi;
fi;

if fini = false then
#Step 3.
prlista3:= lowfactors(a &^ aexp - 1) minus (prlista2 union
prlista1 union factorset(2*b));
if nops(prlista3)> 0 then
for i from 1 to nops(prlista3) do
bexp := lcm(bexp, order(b, prlista3[i]));
od;
fi;
prlistb3:= lowfactors(b &^ bexp - 1) minus (prlistb2 union
prlistb1 union factorset(2*a));
if nops(prlistb3)> 0 then
for i from 1 to nops(prlistb3) while fini = false do
aexp := lcm(aexp, order(a , prlistb3[i]));
if aexp > x2bound then fini := true; end if;
od;
fi;
fi;

if fini = false then
#Step 4.
prlista4:= morefactors(a &^ aexp - 1) minus (prlista3 union
prlista2 union prlista1 union factorset(2*b));
if nops(prlista4)> 0 then
for i from 1 to nops(prlista4) do
bexp := lcm(bexp, order(b, prlista4[i]));
od;
fi;
prlistb4:= morefactors(b &^ bexp - 1) minus (prlistb3 union
prlistb2 union prlistb1 union factorset(2*a));
if nops(prlistb4)> 0 then
for i from 1 to nops(prlistb4) while fini = false do
aexp := lcm(aexp, order(a , prlistb4[i]));
if aexp > x2bound then fini := true; end if;
od;
fi;
fi;

if fini = false then print(a,aexp, morefactors(a&^aexp - 1), b,

```

```

bexp, morefactors(b&^bexp - 1)); end if;
fini;
end;

```

```

bootstrap4fold := proc(a, b)

```

```

local prlista1, prlista2, prlista3, prlista4, prlistb1, prlistb2, prlistb3, prlistb4, i, j, p, aexp,
bexp, x2bound, fini;

```

```

x2bound := evalf(3798 * log(b));

```

```

fini := false;

```

```

aexp := numtheory:-order(a, b);

```

```

bexp := ilcm(a, numtheory:-order(b, a));

```

```

prlista1 := minus(lowestfactors(a &^ aexp - 1), numtheory:-factorset(2 * b));

```

```

if 0 < nops(prlista1) then

```

```

for i to nops(prlista1) do

```

```

    bexp := lcm(bexp, numtheory:-order(b, prlista1[i]))

```

```

end do

```

```

end if;

```

```

prlistb1 := minus(lowestfactors(b &^ bexp - 1), numtheory:-factorset(2 * a));

```

```

if 0 < nops(prlistb1) then

```

```

for i to nops(prlistb1) do

```

```

    aexp := lcm(aexp, numtheory:-order(a, prlistb1[i]));

```

```

    if x2bound < aexp then fini := true end if

```

```

end do

```

```

end if;

```

```

if fini = false then

```

```

    prlista2 := minus(lowerfactors(a &^ aexp - 1), union(prlista1, numtheory:-factorset(2
    * b)));

```

```

if 0 < nops(prlista2) then

```

```

for i to nops(prlista2) do

```

```

    bexp := lcm(bexp, numtheory:-order(b, prlista2[i]))

```

```

end do

```

```

end if;

```

```

prlistb2 := minus(lowerfactors(b &^ bexp - 1), union(prlistb1, numtheory:-factorset(2
* a)));

```

```

if 0 < nops(prlistb2) then

```

```

for i to nops(prlistb2) while fini = false do

```

```

    aexp := lcm(aexp, numtheory:-order(a, prlistb2[i]));

```

```

    if x2bound < aexp then fini := true end if

```

```

end do

```

```

end if

```

```

end if;

```

```

if fini = false then

```

```

    prlista3 := minus(lowerfactors(a &^ aexp - 1), union(union(prlista2, prlista1),

```

(6)

```

numtheory:-factorset(2 * b));
if 0 < nops(prlista3) then
    for i to nops(prlista3) do
        bexp := lcm(bexp, numtheory:-order(b, prlista3[i]));
    end do
end if;
prlistb3 := minus(lowfactors(b &^ bexp - 1), union(union(prlistb2, prlistb1),
numtheory:-factorset(2 * a)));
if 0 < nops(prlistb3) then
    for i to nops(prlistb3) while fini = false do
        aexp := lcm(aexp, numtheory:-order(a, prlistb3[i]));
        if x2bound < aexp then fini := true end if
    end do
end if
if fini = false then
    prlista4 := minus(morefactors(a &^ aexp - 1), union(union(union(prlista3,
prlista2), prlista1), numtheory:-factorset(2 * b)));
    if 0 < nops(prlista4) then
        for i to nops(prlista4) do
            bexp := lcm(bexp, numtheory:-order(b, prlista4[i]));
        end do
    end if;
    prlistb4 := minus(morefactors(b &^ bexp - 1), union(union(union(prlistb3,
prlistb2), prlistb1), numtheory:-factorset(2 * a)));
    if 0 < nops(prlistb4) then
        for i to nops(prlistb4) while fini = false do
            aexp := lcm(aexp, numtheory:-order(a, prlistb4[i]));
            if x2bound < aexp then fini := true end if
        end do
    end if
end if;
if fini = false then
    print(a, aexp, morefactors(a &^ aexp - 1), b, bexp, morefactors(b &^ bexp - 1))
end if;
fini
end proc
> for b from 2 to 100 do
    for a from 2 to evalf(2411.31 * log(b)) do
        if gcd(a, b) = 1 then
            bootstrap4fold(a, b);
        fi;
    end do
end do

```

od;
od:

```
2, 2, {3}, 3, 2, {2}
4, 2, {3, 5}, 3, 4, {2, 5}
2, 4, {3, 5}, 5, 2, {2, 3}
2, 6, {3, 7}, 7, 2, {2, 3}
5807, 2, {2, 3, 11, 2903}, 12, 24460553171, {11, 2903, 5807}
7, 4, {2, 3, 5}, 15, 7, {2, 7}
49, 2, {2, 3, 5}, 15, 49, {2, 7}
17, 2, {2, 3}, 18, 17, {17}
6047, 2, {2, 3, 7, 3023}, 28, 27621202391, {3, 3023, 6047}
2113, 1, {2, 3, 11}, 33, 23243, {2, 2113}
5807, 2, {2, 3, 11, 2903}, 33, 24460553171, {2, 2903, 5807}
7559, 2, {2, 3, 5, 7, 3779}, 36, 53960155829, {5, 7, 3779, 7559}
5749, 2, {2, 3, 5, 23, 479}, 46, 658151269, {3, 5, 479, 5749}
167, 2, {2, 3, 7, 83}, 84, 13861, {83, 167}
10457, 2, {2, 3, 7, 83, 1307}, 84, 8924746247, {83, 1307, 10457} (7)
```

```
>
```

(8)

```
> ifactor(12^11-1);
```

```
(11)^2 (23) (266981089) (9)
```

```
> order(5807, 266981089);
```

$a^x - b^y = c$ bootstrapping sieve a, b composite

15 Dec 2007

22 Dec 2007

24 Dec 2007

29 Dec 2007

We are testing for $a^{x_2} - a^{x_1} = b^{y_2} - b^{y_1}$ for $(a, b) = 1$

with the conditions

$b < 100$,

$a < 2411.31 \log(b)$

Rewrite it as $a^{x_1} (a^{(x_2-x_1)} - 1) = b^{y_1} (b^{(y_2-y_1)} - 1)$

We assume that $x_1 > m$ so we begin at $x_1 = 2$, hence we can assume that a divides $b^{y_2} - b^{y_1}$.

If $a^{x_2-x_1} > 3798 \log(b)$ then we have exceeded our bound and so know there is no solution.

```
> restart; with(numtheory) :
```

```
> ithprime(50); ithprime(1000); ithprime(100000);
```

```
229
```

7919
1299709

(10)

```
> lowestfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i from 1 to 50 do
    if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
  od;
  prlist;
end;
```

lowestfactors := proc(n) (11)

```
  local i, prlist;
  prlist := { };
  for i to 50 do
    if mod(n, ithprime(i)) = 0 then prlist := { ithprime(i), op(prlist) } end if
  end do;
  prlist
end proc
```

```
> lowerfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i from 1 to 1000 do
    if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
  od;
  prlist;
end;
```

lowerfactors := proc(n) (12)

```
  local i, prlist;
  prlist := { };
  for i to 1000 do
    if mod(n, ithprime(i)) = 0 then prlist := { ithprime(i), op(prlist) } end if
  end do;
  prlist
end proc
```

```
> lowfactors := proc(n)
  local i, prlist;
  prlist := { };
  for i from 1 to 10000 do
    if n mod ithprime(i) = 0 then prlist := { op(prlist), ithprime(i) }; fi;
  od;
  prlist;
end;
```

lowfactors := proc(n) (13)

```
  local i, prlist;
```

```

prlist := { };
for i to 10000 do
    if mod(n, ithprime(i)) = 0 then prlist := {ithprime(i), op(prlist)} end if
end do;
prlist
end proc

```

```

>
> morefactors := proc (n)
    local i, prlist;
    prlist := { };
    for i to 100000 do
        if `mod`(n, ithprime(i)) = 0 then
            prlist := {ithprime(i), op(prlist)}
        end if
    end do;
    prlist
end proc;

```

```

morefactors := proc(n)
    local i, prlist;
    prlist := { };
    for i to 100000 do
        if mod(n, ithprime(i)) = 0 then prlist := {ithprime(i), op(prlist)} end if
    end do;
    prlist
end proc

```

(14)

```

>
> bootstrap4fold := proc(a,b)
    local prlista1, prlista2, prlista3, prlista4, prlistb1,
    prlistb2, prlistb3, prlistb4, i, j, p, aexp, bexp, x2bound,
    fini;
    x2bound:= evalf(3798*log(b));
    fini:= false;
    #first step; this is just a repeat of checkaexp and checkbexp.
    aexp := order(a,b); bexp:= ilcm(a, order(b,a)); #note we
    assume a divides bexp
    prlista1:= lowestfactors(a&^aexp - 1) minus factorset(2*b);
    if nops(prlista1)> 0 then
        for i from 1 to nops(prlista1) do
            bexp := lcm(bexp, order(b, prlista1[i]));
        od;
    fi;
    #we assume a divides the exponent of b
    prlistb1:= lowestfactors(b&^bexp - 1) minus factorset(2*a);

```

```

if nops(prlistb1)> 0 then
for i from 1 to nops(prlistb1) do
aexp := lcm(aexp, order(a, prlistb1[i]));
if aexp > x2bound then fini := true; end if;
od;
fi;

if fini = false then
#step 2
prlista2:= lowerfactors(a &^ aexp - 1) minus (prlista1 union
factorset(2*b));
if nops(prlista2)> 0 then
for i from 1 to nops(prlista2) do
bexp := lcm(bexp, order(b, prlista2[i]));
od;
fi;
prlistb2:= lowerfactors(b &^ bexp - 1) minus (prlistb1 union
factorset(2*a));
if nops(prlistb2)> 0 then
for i from 1 to nops(prlistb2) while fini = false do
aexp := lcm(aexp, order(a, prlistb2[i]));
if aexp > x2bound then fini := true; end if;
od;
fi;
fi;

if fini = false then
#Step 3.
prlista3:= lowfactors(a &^ aexp - 1) minus (prlista2 union
prlista1 union factorset(2*b));
if nops(prlista3)> 0 then
for i from 1 to nops(prlista3) do
bexp := lcm(bexp, order(b, prlista3[i]));
od;
fi;
prlistb3:= lowfactors(b &^ bexp - 1) minus (prlistb2 union
prlistb1 union factorset(2*a));
if nops(prlistb3)> 0 then
for i from 1 to nops(prlistb3) while fini = false do
aexp := lcm(aexp, order(a , prlistb3[i]));
if aexp > x2bound then fini := true; end if;
od;

```

```

fi;
fi;

if fini = false then
#Step 4.
prlista4:= morefactors(a &^ aexp - 1) minus (prlista3 union
prlista2 union prlista1 union factorset(2*b));
if nops(prlista4)> 0 then
for i from 1 to nops(prlista4) do
bexp := lcm(bexp, order(b, prlista4[i]));
od;
fi;
prlistb4:= morefactors(b &^ bexp - 1) minus (prlistb3 union
prlistb2 union prlistb1 union factorset(2*a));
if nops(prlistb4)> 0 then
for i from 1 to nops(prlistb4) while fini = false do
aexp := lcm(aexp, order(a , prlistb4[i]));
if aexp > x2bound then fini := true; end if;
od;
fi;
fi;

if fini = false then print(a,aexp, morefactors(a&^aexp - 1), b,
bexp, morefactors(b&^bexp - 1)); end if;
fini;
end;

```

bootstrap4fold := **proc**(*a*, *b*)

(15)

```

local prlista1, prlista2, prlista3, prlista4, prlistb1, prlistb2, prlistb3, prlistb4, i, j, p, aexp,
bexp, x2bound, fini;
x2bound := evalf(3798*log(b));
fini := false;
aexp := numtheory:-order(a, b);
bexp := ilcm(a, numtheory:-order(b, a));
prlista1 := minus(lowestfactors(a &^ aexp - 1), numtheory:-factorset(2*b));
if 0 < nops(prlista1) then
    for i to nops(prlista1) do
        bexp := lcm(bexp, numtheory:-order(b, prlista1[i]))
    end do
end if;
prlistb1 := minus(lowestfactors(b &^ bexp - 1), numtheory:-factorset(2*a));
if 0 < nops(prlistb1) then
    for i to nops(prlistb1) do

```

```

    aexp := lcm(aexp, numtheory:-order(a, prlistb1[i]));
    if x2bound < aexp then fini := true end if
end do
end if;
if fini = false then
    prlista2 := minus(lowerfactors(a &^ aexp - 1), union(prlista1, numtheory:-factorset(2
    * b)));
    if 0 < nops(prlista2) then
        for i to nops(prlista2) do
            bexp := lcm(bexp, numtheory:-order(b, prlista2[i]))
        end do
    end if;
    prlistb2 := minus(lowerfactors(b &^ bexp - 1), union(prlistb1, numtheory:-factorset(2
    * a)));
    if 0 < nops(prlistb2) then
        for i to nops(prlistb2) while fini = false do
            aexp := lcm(aexp, numtheory:-order(a, prlistb2[i]));
            if x2bound < aexp then fini := true end if
        end do
    end if
end if;
if fini = false then
    prlista3 := minus(lowfactors(a &^ aexp - 1), union(union(prlista2, prlista1),
    numtheory:-factorset(2 * b)));
    if 0 < nops(prlista3) then
        for i to nops(prlista3) do
            bexp := lcm(bexp, numtheory:-order(b, prlista3[i]))
        end do
    end if;
    prlistb3 := minus(lowfactors(b &^ bexp - 1), union(union(prlistb2, prlistb1),
    numtheory:-factorset(2 * a)));
    if 0 < nops(prlistb3) then
        for i to nops(prlistb3) while fini = false do
            aexp := lcm(aexp, numtheory:-order(a, prlistb3[i]));
            if x2bound < aexp then fini := true end if
        end do
    end if
end if;
if fini = false then
    prlista4 := minus(morefactors(a &^ aexp - 1), union(union(union(prlista3,
    prlista2), prlista1), numtheory:-factorset(2 * b)));

```

```

if 0 < nops(prlista4) then
  for i to nops(prlista4) do
    bexp := lcm(bexp, numtheory:-order(b, prlista4[i]))
  end do
end if;
prlistb4 := minus(morefactors(b &^ bexp - 1), union(union(union(prlistb3,
prlistb2), prlistb1), numtheory:-factorset(2 * a)));
if 0 < nops(prlistb4) then
  for i to nops(prlistb4) while fini = false do
    aexp := lcm(aexp, numtheory:-order(a, prlistb4[i]));
    if x2bound < aexp then fini := true end if
  end do
end if
if fini = false then
  print(a, aexp, morefactors(a &^ aexp - 1), b, bexp, morefactors(b &^ bexp - 1))
end if;
  fini
end proc

```

```

> for b from 2 to 100 do
  for a from 2 to evalf(2411.31 * log(b)) do
    if gcd(a, b) = 1 then
      bootstrap4fold(a, b);
    fi;
  od;
od;

```

2, 2, {3}, 3, 2, {2}
 4, 2, {3, 5}, 3, 4, {2, 5}
 2, 4, {3, 5}, 5, 2, {2, 3}
 2, 6, {3, 7}, 7, 2, {2, 3}
 5807, 2, {2, 3, 11, 2903}, 12, 24460553171, {11, 2903, 5807}
 7, 4, {2, 3, 5}, 15, 7, {2, 7}
 49, 2, {2, 3, 5}, 15, 49, {2, 7}
 17, 2, {2, 3}, 18, 17, {17}
 6047, 2, {2, 3, 7, 3023}, 28, 27621202391, {3, 3023, 6047}
 2113, 1, {2, 3, 11}, 33, 23243, {2, 2113}
 5807, 2, {2, 3, 11, 2903}, 33, 24460553171, {2, 2903, 5807}
 7559, 2, {2, 3, 5, 7, 3779}, 36, 53960155829, {5, 7, 3779, 7559}
 5749, 2, {2, 3, 5, 23, 479}, 46, 658151269, {3, 5, 479, 5749}
 167, 2, {2, 3, 7, 83}, 84, 13861, {83, 167}
 10457, 2, {2, 3, 7, 83, 1307}, 84, 8924746247, {83, 1307, 10457}

(16)

We now factor these cases to find large factors that will exceed the bound.

```

> 5807, 2, {2, 3, 11, 2903}, 12, 24460553171, {11, 2903, 5807}
> ifactor(24460553171);
(1451) (5807) (2903) (17)
(18)
> ifactor(12^1451 - 1, easy);
(11) _c1553_1 (283397713) (2903) (19)
> order(5807, 283397713);
47232952 (20)
> (283397713 - 1) / %;
6 (21)
>
> 7, 4, {2, 3, 5}, 15, 7, {2, 7} (22)
> ifactor(7^4 - 1); ifactor(15^7 - 1);
(2)^5 (3) (5)^2
(2) (7)^2 (1743463) (23)
> order(7, 1743463);
1743462 (24)
>
> 17, 2, {2, 3}, 18, 17, {17}
> ifactor(17^2 - 1); ifactor(18^17 - 1);
(2)^5 (3)^2
(17)^2 (7563707819165039903) (25)
> order(17, 7563707819165039903);
3781853909582519951 (26)
>
> 6047, 2, {2, 3, 7, 3023}, 28, 27621202391, {3, 3023, 6047}
> ifactor(27621202391);
(1511) (6047) (3023) (27)
> ifactor(28^1511 - 1, easy);
(3)^3 _c2176_1 (2070071) (3023) (28)
> order(6047, 2070071);
1035035 (29)
>
> 2113, 1, {2, 3, 11}, 33, 23243, {2, 2113}
> ifactor(23243);
(11) (2113) (30)
> ifactor(33^11 - 1);
(2)^5 (747487377451) (2113) (31)
> order(2113, 747487377451);
(32)

```

```

124581229575 (32)
>
> 5807, 2, {2, 3, 11, 2903}, 33, 24460553171, {2, 2903, 5807}
> ifactor(24460553171);
(1451) (5807) (2903) (33)
> ifactor(331451 - 1, easy);
(2)5 _c2192_1 (4834733) (2903) (34)
> order(5807, 4834733);
690676 (35)
>
> 7559, 2, {2, 3, 5, 7, 3779}, 36, 53960155829, {5, 7, 3779, 7559}
> ifactor(53960155829);
(7559) (3779) (1889) (36)
> ifactor(367559 - 1, easy);
(5) (7) _c11763_1 (37)
No factors are found for any of the small exponents.
Fortunately, since 36 is a square, we do not need to deal with this case.
>
> 5749, 2, {2, 3, 5, 23, 479}, 46, 658151269, {3, 5, 479, 5749}
> ifactor(658151269);
(239) (479) (5749) (38)
>  $\frac{5748}{479}$ ;
12 (39)
> ifactor(46239 - 1, easy);
(3)2 (5) (479) _c394_1 (40)
> ifactor(46479 - 1, easy);
(3)2 (5) _c785_1 (2092273) (5749) (41)
> order(5749, 2092273);
523068 (42)
>
> 167, 2, {2, 3, 7, 83}, 84, 13861, {83, 167}
> ifactor(13861);
(83) (167) (43)
> ifactor(8483 - 1, easy);
(83)2 (167) _c154_1 (44)
> ifactor(84167 - 1, easy);
(83) _c304_1 (4128812950503781) (45)
> order(167, 4128812950503781);
344067745875315 (46)
>

```

```
[> 10457, 2, {2, 3, 7, 83, 1307}, 84, 8924746247, {83, 1307, 10457}
[> ifactor(8924746247);
                                     (653) (1307) (10457)                                (47)
```

```
[> ifactor(84653 - 1, easy);
                                     (83) (1307) _c1238_1 (314024357080123)          (48)
```

```
[> order(10457, 314024357080123);
                                     314024357080122                                (49)
```

```
[>
```

[So we have shown that each exceptional case is handled.