Compute a sqrt in **Z** using a linear p-adic Newton iteration.

```
> NI := proc(u0::integer,a::posint,p::prime,B::posint) local u,e,uk,
  pk,k,i;
      u := mods(u0,p);
      i := (2*u0)^(-1) mod p;
      pk := p;
      k := 1;
      while true do
          e := a-u^2;
          if e=0 then return u; fi;
          if pk > 2*B then return FAIL; fi;
          e := iquo(e,pk) mod p;
          uk := mods(i*e,p);
          u := u + uk*pk;
          pk := p*pk;
      od;
  end:
> a := 131^2;
```

$$a := 17161$$

```
> p := 7;
```

$$p := 7$$

```
> Factor( x^2-a ) mod p;
```

$$(x+5)\,(x+2)$$

```
> NI(-2,a,p,200);
```

$$131$$

```
> NI(2,a,p,200);
```

$$-131$$

```
> p := prevprime(10^4):
> a := 3^20000: u0 := 3^10000 mod p:
> time(NI(u0,a,p,a));
```

$$0.109$$

```
> a := a*a: u0 := u0*u0 mod p:
> time(NI(u0,a,p,a));
```

$$0.380$$

```
> a := a*a: u0 := u0*u0 mod p:
> time(NI(u0,a,p,a));
```

$$2.300$$

```
> a := a*a: u0 := u0*u0 mod p:
> time(NI(u0,a,p,a));
```

$$11.452$$