```
mod_bounds  THM

⊢  ∀a:ℤ, n:ℕ+. 0≤(a mod n) & (a mod n)<n  by GenUnivCD
|
|\
| 1. a : ℤ
| 2. n : ℕ+
| ⊢  0≤(a mod n)
| | by Inst Thm* ∀a:ℤ, n:ℕ+. (a mod n) ∈ ℕ    {modulus_wf} [a;n]
| |
| 3. (a mod n) ∈ ℕ
| ⊢  0≤(a mod n)  by Analyze 3
| |
| 3. (a mod n) ∈ ℤ
| 4. (a mod n)≥0
| ⊢  0≤(a mod n)  by Auto
|
.\
. 1. a : ℤ
. 2. n : ℕ+
. ⊢  (a mod n)<n  by Unfold ‘modulus‘ 0
. |
. ⊢  if 0≤₂a→a rem n ;((-a) rem n)=₂0→0 else n-((-a) rem n) fi<n
. | by (BoolCasesOnCExp (0≤₂a)) THEN (RWH ifthenelse_evalC 0)
. |
. |\
. | 3. 0≤a
. | ⊢  (a rem n)<n
. |   by BackThru
. |      Thm* ∀a:ℕ, n:ℕ+. 0≤(a rem n) & (a rem n)<n    {rem_bounds_1}
. |
. .\
. . 3. a<0
. . ⊢  if ((-a) rem n)=₂0→ 0 else n-((-a) rem n) fi<n
. . | by (BoolCasesOnCExp (((-a) rem n)=₂0))
. . |    THEN
. . |    (RWH ifthenelse_evalC 0)
. . |
. . |\
. . | 4. ((-a) rem n) = 0
. . | ⊢  0<n  by Auto
. . |
. . .\
. . . 4. ¬((-a) rem n) = 0
. . . ⊢  n-((-a) rem n)<n
. . . | by (Using [‘n’,(((-a) rem n)-n)]
. . . |     (BackThru
. . . |      Thm* ∀a,b,n:ℤ. a+n<b+n ⇒ a<b    {add_cancel_in_lt}))
. . . |    THEN
. . . |    (ArithSimp 0)
. . . |
. . . ⊢  0<((-a) rem n)  by Assert (0≤((-a) rem n))
. . . |
. . . |\
. . . | .....assertion .....
```

```
. . . | ⊢  0≤((-a) rem n)
. . . |   by BackThru
. . . |      Thm* ∀a:ℕ, n:ℕ+. 0≤(a rem n) & (a rem n)<n
. . . |      {rem_bounds_1}
. . . |
. . . .\
. . . . 5. 0≤((-a) rem n)
. . . . ⊢  0<((-a) rem n)  by Auto
END
```