```
.Pred  esPrimo(n: ℤ) {
       n > 1 ∧
       (∀m: ℤ)(1 < m < n →_L n mod m ≠ 0)
}

.Pred  esMayorPrimoQueDivide(x: ℤ, y: ℤ) {
       esPrimo(y) ∧ divide(y, x) ∧
       (∀n: ℤ)(esPrimo(n) ∧ divide(n, x) → n ≤ y)
}

.Pred  pertenece(e: ℤ, s: seq<ℤ>) {
       (∃i: ℤ)(0 ≤ i < |s| ∧_L s[i] = e)
}

.Pred  esPrefijo(p: seq<char>, s: seq<char>) {
       |p| ≤ |s| ∧_L
         (∀i: ℤ)(0 ≤ i < |p| →_L p[i] = s[i])
}

.Pred  todosLosPrimosEnPosicionesPares(s: seq<ℤ>) {
       (∀i: ℤ)((0 ≤ i < |s| ∧ esPrimo(s[i])) → i mod 2 = 0)
}                                              divide(2, i)

.Pred  tieneUnPrimoQueDivideAlResto(s: seq<ℤ>) {
       (∃e: ℤ)(pertenece(e, s) ∧ esPrimo(e) ∧
            divideATodos(e, s)
       }

Pred   divideATodos(e: ℤ, s: seq<ℤ>) {
       (∀i: ℤ)(0 ≤ i < |s| →_L divide(e, s[i]))
       }
```

• aux obtenerMayor(t: tupla⟨ℤ, ℤ⟩): ℤ =
IfThenElse($t_0 > t_1$, t_0, t_1)

• aux dígitoMenosSignificativo(x: ℤ): ℤ = $x \bmod 10$

• aux cantidadApariciones(e: ℤ, s: seq⟨ℤ⟩): ℤ =
$\sum_{i=0}^{|s|-1}$ IfThenElse(e = s[i], 1, 0)

• aux sumaPosicionesPares(s: seq⟨ℤ⟩): ℤ =
$\sum_{i=0}^{|s|-1}$ IfThenElse(i mod 2 = 0, s[i], 0)

• Proc factorizaciónEnPrimos(in n: ℤ): seq⟨tupla⟨ℤ, ℤ⟩⟩ {
requiere {n > 0}
asegura {n = 1 → res = ⟨⟩ ∧
n ≠ 1 → (sonTodosLosFactores(n, res) ∧
ordenadaPorPrimos(res))
}
}

pred sonTodosLosFactores(n: ℤ, r: seq⟨tupla⟨ℤ, ℤ⟩⟩) {
todoPEsPrimo(r) ∧ sonExponentesCorrectos(n, r)
∧ todosLosFactores(n, r)
}

pred todoPEsPrimo(r: seq⟨tupla⟨ℤ, ℤ⟩⟩) {
$(\forall i: ℤ)(0 \leq i < |r| \rightarrow_L esPrimo(r[i]_0))$
}

pred sonExponentesCorrectos(n: ℤ, r: seq⟨tupla⟨ℤ, ℤ⟩⟩) {
$(\forall i: ℤ)(0 \leq i < |r| \rightarrow_L$
$(r[i]_1 > 0 \wedge_L (divide(r[i]_0^{r[i]_1}, n) \wedge \neg divide(r[i]_0^{(r[i]_1 + 1)}, n))))$
}

```
pred todosLosFactores(n: ℤ, r: seq<tupla2<ℤ, ℤ>>) {
    (∀p: ℤ)(esPrimo(p) ∧ divide(p, n) →
        (∃i: ℤ)(0 ≤ i < |r| ∧_L r[i]_0 = p))
}
pred ordenadaPorPrimos(r: seq<tupla2<ℤ, ℤ>>) {
    (∀i: ℤ)(0 ≤ i < |r| - 1 →_L r[i]_0 < r[i+1]_0)
}
```