\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod NATURAL is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

sort N .

sort NoN .

subsort NoN < N .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

op cero : -> N .

op sucesor : N -> N .

op suma : N N -> N .

op esCero : N -> Bool .

op esIgual : N N -> Bool .

op esDistinto : N N -> Bool .

op NODEFINIDO : -> NoN .

op INFINITO : -> NoN .

op NEGATIVO : -> NoN .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var n m : N .

eq suma(cero, n) = n .

eq suma(sucesor(m), n) = sucesor(suma(m, n)) .

eq esCero(cero) = true .

eq esCero(sucesor(n)) = false .

eq esIgual(cero, n) = esCero(n) .

eq esIgual(sucesor(n), cero) = false .

eq esIgual(sucesor(n), sucesor(m)) = esIgual(n, m) .

eq esDistinto(n, m) = not esIgual(n, m) .

endfm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod VOCAL is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

sort V .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ops A E I O U : -> V .

op esIgual : V V -> Bool .

op esDistinta : V V -> Bool .

op esMenor : V V -> Bool .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var v w : V .

eq esIgual(v, v) = true .

eq esIgual(v, w) = false .

eq esDistinta(v, w) = not esIgual(v, w) .

eq esMenor(v, v) = false .

eq esMenor(A, E) = true .

eq esMenor(A, I) = true .

eq esMenor(A, O) = true .

eq esMenor(A, U) = true .

eq esMenor(E, I) = true .

eq esMenor(E, O) = true .

eq esMenor(E, U) = true .

eq esMenor(I, O) = true .

eq esMenor(I, U) = true .

eq esMenor(O, U) = true .

eq esMenor(U, O) = false .

eq esMenor(U, I) = false .

eq esMenor(U, E) = false .

eq esMenor(U, A) = false .

eq esMenor(O, I) = false .

eq esMenor(O, E) = false .

eq esMenor(O, A) = false .

eq esMenor(I, E) = false .

eq esMenor(I, A) = false .

eq esMenor(E, A) = false .

endfm

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOMBRE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

fmod PILA is

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CONJUNTOS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

protecting BOOL .

protecting VOCAL .

protecting NATURAL .

sort MensajePilas .

sort P .

subsorts MensajePilas < V .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SINTAXIS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

op pilaVacia : -> P .

op esVacia : P -> Bool .

op push : V P -> P .

op pop : P -> P .

op tope : P -> V .

op esIgual : P P -> Bool .

op primero : P -> V .

op tamano : P -> N .

op cuentaVocal : V P -> N .

op ERRORPILAVACIA : -> MensajePilas .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* SEMANTICA \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

var p q : P .

var v w t r : V .

eq esVacia(pilaVacia) = true .

eq esVacia(push(v, p)) = false .

eq pop(pilaVacia) = pilaVacia .

eq pop(push(v, p)) = p .

eq tope(pilaVacia) = ERRORPILAVACIA .

eq tope(push(v, p)) = v .

eq esIgual(pilaVacia, pilaVacia) = true .

eq esIgual(p, pilaVacia) = false .

eq esIgual(pilaVacia, p) = false .

eq esIgual(pilaVacia, pop(push(v, p))) = true .

eq esIgual(push(v, p), pilaVacia) = false .

eq esIgual(push(v, p), push(v, p)) = true .

eq esIgual(push(v, p), pop(push(v, p))) = false .

eq esIgual(push(v, p), pop(push(v, push(v, p)))) = true .

eq esIgual(push(A, p), push(E, p)) = false .

eq esIgual(push(A, p), push(I, p)) = false .

eq esIgual(push(A, p), push(O, p)) = false .

eq esIgual(push(A, p), push(U, p)) = false .

eq esIgual(push(E, p), push(I, p)) = false .

eq esIgual(push(E, p), push(O, p)) = false .

eq esIgual(push(E, p), push(U, p)) = false .

eq esIgual(push(I, p), push(O, p)) = false .

eq esIgual(push(I, p), push(U, p)) = false .

eq esIgual(push(v, p), push(t, q)) = if esIgual(v, t) == true then esIgual(p, q) else false fi .

eq primero(pilaVacia) = ERRORPILAVACIA .

eq primero(push(v, pilaVacia)) = v .

eq primero(push(v, p)) = primero(p) .

eq primero(push(v, push(A, p))) = A .

eq primero(pop(push(v, push(A, p)))) = A .

eq tamano(pilaVacia) = cero .

eq tamano(push(v, p)) = sucesor(tamano(p)) .

eq tamano(push(v, push(v, p))) = sucesor(sucesor(cero)) .

eq tamano(push(v, pop(push(v, p)))) = sucesor(cero) .

eq cuentaVocal(v, pilaVacia) = cero .

eq cuentaVocal(v, push(t, p)) = if esIgual(v, t) == true then sucesor(cuentaVocal(v, p)) else cuentaVocal(v, p) fi .

endfm