Acción insertarConOrden (dato r E Tdata, dato nuevo E Tpersona)

Léxico local

i, j E N

encontrado E B

Inicio

encontrado <-- Falso

si NO(llena(r)) entonces

i <-- r.cant

mientras (i >= 0 y (r.a[i].apellido > nuevo.apellido) y NO(encontrado)) hacer

si (r.a[i].apellido = nuevo.apellido) entonces

// Si los apellidos son iguales, compara por nombre y DNI

si ((r.a[i].nombre> nuevo.nombre) ) o

(r.a[i].nombre = nuevo.nombre y r.a[i].dni > nuevo.dni) entonces

encontrado <-- Verdadero

fsi

fsi

si NO(encontrado) entonces

i <-- i - 1

fsi

Fmientras

// En este punto, i apunta al índice donde debe insertarse la nueva persona.

// Necesitamos mover hacia delante elementos mayores a ese índice en el registro.

para (j <-- r.cant, j > i, j <-- j - 1) hacer

r.a[j + 1] <-- r.a[j]

fpara

r.a[i + 1] <-- nuevo

r.cant <-- r.cant + 1

salida: mje “Persona insertada correctamente a la lista”

sino

salida: mje “La lista está llena”

fsi

facción

Acción eliminarPersona (dato s E Tdata)

Léxico local

dniAEliminar E Entero

i, j E Entero

mje E cadena

Inicio

mje: "Ingrese el DNI de la persona que desea eliminar: "

Salida: mje

Entrada: dniAEliminar

i <-- 0

mientras (i < s.cant y s.a[i].dni ≠ dniAEliminar) hacer

i <-- i + 1

Fmientras //me muevo hasta encontrar la posición perteneciente al dni ingresado

si (i < s.cant) entonces

// Se encontró la persona con el DNI a eliminar

para (j <-- i; j < s.cant - 1; j <-- j + 1) hacer

s.a[j] <-- s.a[j + 1]

fpara // movemos todos los registros con posición superior al registro eliminado

//una posición hacia atrás.

s.cant <-- s.cant – 1 // resto en uno la cantidad de registros

salida: mje "Persona con DNI ", dniAEliminar, " eliminada correctamente de la lista"

sino

salida: mje "Persona con DNI ", dniAEliminar, " no encontrada en la lista"

fsi

facción