

## EJERCICIO 10: REPRESENTAR LOS MIEMBROS DE UNA FAMILIA

Se ha formado una tabla familias con 1000 componentes numerados desde 1.

### 1. Definir el tipo PERSONA

```
tipo PERSONA

estructura
  identidad : IDENTIDAD
  edad : ENTERO
  nump : ENTERO #número de padres
  progenitores : TABLA[PERSONA][0,nump] # lista personas con las
que tiene relación de parentesco

invariantes
  edad ≥ 0
  0 ≤ nump ≤ 2 #puede tener 0,1 o 2 progenitores

fin tipo
```

### Declarar tabla familia

Cuando una persona registrada en la tabla no tiene padre o madre registrado, el atributo correspondiente es HUÉRFANO.

```
# inicializamos la tabla familia
variable
  familia : TABLA[PERSONA][1,1000]
  i : ENTERO # indice en la tabla 'familia'

inicialización
  i ← 1

mientras
  i ≤ 1000

# bucle para asignar atributo huerfano
repetir
  si
    familia[i].nump → 0
    entonces
      familia[i].nump ← 'HUERFANO'
      i+1 ← i

  si no
    i+1 ← i

  fin si
fin repetir
```

Cuando una celda no está ocupada porque la persona que contenía se ha borrado, su identificador es BORRADO. Las celdas que nunca han recibido valor, cuando hay menos de 1000 personas registradas, tienen un identificador igual a VACÍO.

**2. Dar la lista de todas las personas registradas con una edad de 20 a 30 años.**

```
variables
  familia : TABLA[PERSONA][1,1000]
  i : ENTERO # indice en la tabla 'familia'

inicialización
  i ← 1

mientras
  familia[i] ≠ VACIO # sea una celda inicializada

repetir
  si
    20 ≤ familia[i].edad ≤ 30
    entonces
      escribir(PANTALLA, familia[i]) #la sacamos por pantalla
      i+1 ← i

  si no
    i+1 ← i # seguimos recorriendo la tabla

  fin si
fin repetir
```

**3. Envejecer 1 año a todas las personas registradas.**

```
variables
  familia : TABLA[PERSONA][1,1000]
  i : ENTERO # indice en la tabla 'familia'

inicialización
  i ← 1

mientras
  familia[i] ≠ VACIO # sea una celda inicializada

repetir
  si
    i ← BORRADO # es una casilla borrada
    entonces
      i+1 ← i # la saltamos

  si no
    familia[i].edad ← familia[i].edad + 1
    i+1 ← i

  fin si
fin repetir
```

**4. Establecer la lista de todos los huérfanos de menos de 15 años.**

```
variables
  familia : TABLA[PERSONA][1,1000]
  i : ENTERO # indice en la tabla 'familia'

inicialización
  i ← 1

mientras
  familia[i] ≠ VACIO # sea una celda inicializada

repetir
  si
    familia[i].edad ≤ 15 y familia[i].nump ← 'HUERFANO'
    entonces
      escribir(PANTALLA, familia[i]) #lo sacamos por pantalla
      i+1 ← i
  si no
    i+1 ← i # seguimos recorriendo la tabla

  fin si
fin repetir
```

5. Hacer un algoritmo que determina la identidad del padre de 'Jaime MARTÍN'.

**Algoritmo identidad\_padres**

**Entrada**

familia : TABLA[PERSONA][1,1000]  
n : ENTERO # identificador celda Jaime Martin

**Salida** : t : TABLA[PERSONAS][1,2]

**Precondicion**

familia[n].identidad = 'Jaime Martin'

**Variable**

i : ENTERO  
t : TABLA[PERSONAS][1,2] # donde vamos a meter a los  
progenitores de Jaime (no inicializada)

**Realizacion**

**inicialización**

i ← 1 # vamos a recorrer la tabla en orden

**mientras**

familia[i] ≠ VACIO # sea una celda inicializada

**repetir**

si VERDAD

familia[i] pertenece familia[n].progenitores # si la persona  
pertenece a la lista de progenitores de Jaime

entonces

item(t, familia[n].progenitores) # guardamos

si FALSO

i+1 ← i # continuamos buscando en la tabla

fin si

fin repetir

**Postcondicion**

$\forall k, \text{indice\_min}(t) \leq k \leq \text{indice\_max}(t) \quad t[k] \text{ pertenece}$   
familia[n].progenitores

**fin identidad\_padres**