

### 1ra Evaluación II Término 2007-2009. Diciembre 04, 2007

**Tema 3** (40 puntos) El instituto Ecuatoriano de Seguridad Social (IESS) ofrece un **préstamo** a los afiliados, el que es garantizado con los valores depositados en “**fondos de reserva**” y “**cesantías**”.

Con el objetivo de conocer el número de afiliados que están en capacidad de realizar los préstamos, se solicitó un listado de número de cédula de afiliados al departamento “fondos de reserva” y otro al departamento “cesantía” debido a que operan todavía de manera independiente.

i	reserva[i]	j	cesantía[j]	i	cumple[i]
1	987	1	876	1	0
2	876	2	765	2	1
3	765			3	1
4	654			4	0
...	...			...	
n		m		n	0

Realice un algoritmo que muestre la cantidad de afiliados que cumplen con las condiciones para el préstamo (se encuentran en los dos listados) y luego muestre los números de cédula encontrados.

*Nota: Existen n afiliados con “fondos de reserva” y m afiliados con valores de “cesantía”.*

*Rubrica: Ingreso de cédulas en listas separadas (5 puntos). Selección (10 puntos) y búsqueda de elementos repetidos (15 puntos). Mostrar solo cédulas válidas (10 puntos).*

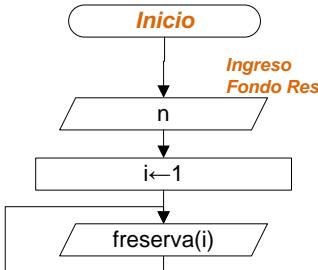
#### Propuesta de Solución:

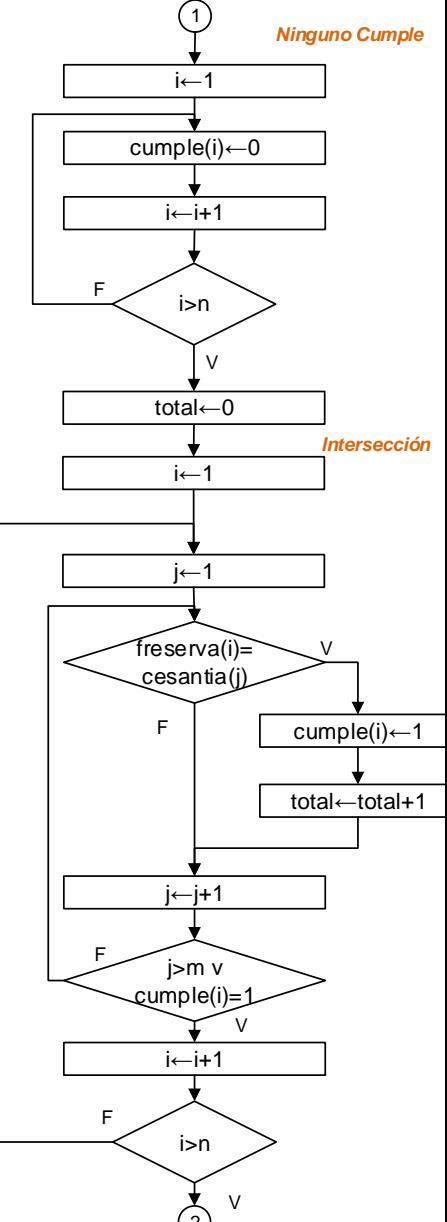
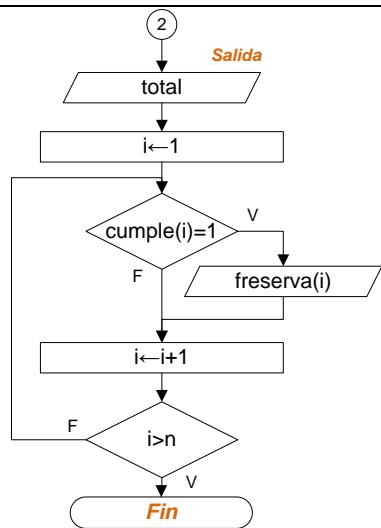
Para el ejemplo, por simplicidad se usan cédulas de 3 dígitos; también se supone que no existen elementos repetidos dentro de cada arreglo. En la nota se indica que las listas de las cédulas de los afiliados tienen tamaño diferente; además el ingreso de datos es separado por tener origen de datos diferentes.

Al inicio del algoritmo, se supondrá que ningún afiliado cumple con los requisitos, por lo que se usa un arreglo de banderas de cumple o no cumple (1 ó 0).

Realizar una búsqueda ordenada de números de cédula: para cada cédula de  $reserva[i]$ ; realizar una búsqueda en el arreglo de  $cesantía[j]$  para los  $m$  elementos. Repetir el proceso, marcando el arreglo de banderas cada vez que se encuentra un valor repetido entre ambas listas.

*Tarea: Realizar el ejercicio, construyendo un tercer arreglo con solamente los elementos que cumplen con la condición.*

Descripción	DIAGRAMA DE FLUJO	Python
Inicio		# ICM00794-Fundam. Computación - FCNM-ESPOL # 1ra Evaluación II Término 2007 # Tema 2. IESS Intersección de listados # propuesta: edelros@espol.edu.ec
Número de afiliados con fondo de Reserva		import numpy as np
Primer Afiliado		n=int(input('Afiliados con Fondo de Reserva: '))
Repita Cedula registrada		freserva=np.zeros(n+1,dtype=int) i=1 while not(i>n): print(i) freserva[i]=input('ingrese cedula: ')
Siguiente afiliado		i=i+1
Hasta completar todos los afiliados		
Número de afiliados con cesantías		m=int(input('Afiliados con Cesantia: '))
Primer afiliado		cesantia=np.zeros(m+1,dtype=int)
...		j=1 while not(j>m):
...		print(j) cesantia[j]=input('ingrese cedula: ')
		j=j+1

<p>Se supone que ninguno cumple</p> <p>Inicia banderas para cada afiliado</p> <p>Realiza intersección</p> <p>Contador de elementos</p> <p>Primero de freserva</p> <p>Repita</p> <p>Primero Cesantía</p> <p>Repita</p> <p>Si las cédulas coinciden</p> <p>Marca la bandera cumple</p> <p>Cuenta un elemento</p> <p>Siguiente de cesantía</p> <p>Hasta comparar con toda la lista de cesantía ó encontró una coincidencia</p> <p>Siguiente de fondos</p> <p>Hasta revisar todos los de fondos</p>		<p># Hipótesis: Ninguno cumple requisito</p> <p>cumple=np.zeros(n+1,dtype=int)</p> <p># Revisa listado buscando iguales</p> <p>total=0</p> <p>i=1</p> <p>while not(i&gt;n):</p> <p>    j=1</p> <p>    while not(j&gt;m or cumple[i]==1):</p> <p>        if freserva[i]==cesantia[j]:</p> <p>            cumple[i]=1</p> <p>            total=total+1</p> <p>        j=j+1</p> <p>    i=i+1</p>
<p>Muestra el número de elementos encontrados</p> <p>Muestra las cédulas solo si se encuentran en los dos listados.</p> <p>...</p>		<p># Muestra cedulas de los que cumplen requisito</p> <p>print('Cumplen requisitos:')</p> <p>print(total)</p> <p>print('Listado: ')</p> <p>i=1</p> <p>while not(i&gt;n):</p> <p>    if (cumple[i]==1):</p> <p>        print(freserva[i])</p> <p>    i=i+1</p> <p>Fin</p>

### Ejecución del algoritmo: iessprestamo.py

```
>>>  
Afiliados con Fondo de Reserva: 3  
1  
ingrese cedula:123  
2  
ingrese cedula:234  
3  
ingrese cedula:345  
Afiliados con Cesantía: 2  
1  
ingrese cedula:655  
2  
ingrese cedula:234
```

Cumplen requisitos:  
1  
Listado:  
234

Nota para desarrollo: Si el tema se usa como proyecto, se deberá incluir el procedimiento para eliminar elementos repetidos por cada lista.