

3ra Evaluación I Término 2009-2010. Septiembre 15, 2009

Tema 2 (20 puntos). Un cajero automático requiere de una función billetes(cantidad) que reciba una cantidad de dólares (número entero) y encuentre su equivalente, usando la menor cantidad de billetes de 50, 20, 10, 5, 1, entregando el resultado en un vector. Ejemplo:

>> billetes(77) ans= 1, 1, 0, 1, 2

Rúbrica: Definir correctamente la función (5 puntos), vector de equivalentes (10 puntos), resultados (5 puntos)

Propuesta de Solución:

Para mostrar la solución primero se desarrollará un algoritmo completo en diagrama de flujo, que es lo que se requiere observar en la primera semana del curso. Para la segunda parte del curso, se usa la propuesta de solución para crear la función solicitada en la 3ra evaluación. El algoritmo se describe como:

- Se solicita la **cantidad** de dinero que se quiere retirar del cajero
- La siguiente operación consiste en determinar cuántos billetes de 50 se entregan, por lo que se usa cociente de cantidad/50.
- Luego se calcula el **saldo** por entregar
- Continuando con operaciones similares para los billetes de 20, 10, 5 y 1 y los muestra al final.

Como algoritmo estructurado:

Descripción	DIAGRAMA DE FLUJO	Python
Ingresar cantidad		<pre># ICM00794-Fundamentos de Computación - FCNM-ESPOL # 3ra Evaluación 2009. Tema 2. Billetes de cajero automático # Propuesta en forma de programa. edelros@espol.edu.ec cantidad=int(input('Cantidad solicitada: '))</pre>
Calcula billetes 50		<pre>#Procedimiento # division entera o cociente // billete50=cantidad//50</pre>
Calcula el saldo		<pre>saldo=cantidad-50*(billete50)</pre>
Continúan operaciones...		<pre>billete20=saldo//20</pre>
		<pre>saldo=saldo-20*(billete20)</pre>
		<pre>billete10=saldo//10</pre>
		<pre>saldo=saldo-10*(billete10)</pre>
		<pre>billete05=saldo//5</pre>
		<pre>billete01=saldo-5*(billete05)</pre>
Muestra resultados		<pre># Salida print ('Billetes de 50:',billete50) print ('Billetes de 20:',billete20) print ('Billetes de 10:',billete10) print ('Billetes de 05:',billete05) print ('Billetes de 01:',billete01)</pre>

Ejecución del algoritmo: billete.py

Cantidad solicitada: 77 Billetes de 50: 1 Billetes de 20: 1 Billetes de 10: 0 Billetes de 05: 1 Billetes de 01: 2 >>>	Cantidad solicitada: 187 Billetes de 50: 3 Billetes de 20: 1 Billetes de 10: 1 Billetes de 05: 1 Billetes de 01: 2 >>>
---	--

Para crear la función del problema, se usa solo el bloque de procedimiento, resumiendo en la línea de función las variables de ingreso y devolviendo las variables de salida. En este caso, para retornar una sola variable se usa un vector. Para usar funciones numéricas de vectores se importa el módulo numpy.

Descripción	Diagrama de Flujo	Python
usará las funciones numéricas Define la función Realiza el procedimiento anterior Asigna los valores al arreglo de salida	<pre> def billetes(cantidad) Procedimiento billete50 ← entero(cantidad/50) saldo ← cantidad-50*(billete50) billete20 ← entero(saldo/20) saldo ← saldo-20*(billete20) billete10 ← entero(saldo/10) saldo ← saldo-10*(billete10) billete05 ← entero(saldo/5) billete01 ← saldo-5*(billete05) entrega(1) ← billete50 entrega(2) ← billete20 entrega(3) ← billete10 entrega(4) ← billete05 entrega(5) ← billete01 return entrega </pre>	<pre> # ICM00794-Fundamentos de Computación - FCNM-ESPOL # 3ra Evaluación 2009. Tema 2. Billetes de cajero automático # Propuesta función. edelros@espol.edu.ec import numpy def billete(cantidad): #Procedimiento # división entera o cociente // billete50=cantidad//50 saldo=cantidad-50*(billete50) billete20=saldo//20 saldo=saldo-20*(billete20) billete10=saldo//10 saldo=saldo-10*(billete10) billete05=saldo//5 billete01=saldo-5*(billete05) # define variable de salida entrega=numpy.array([0, 0, 0, 0, 0], int) entrega[0]=billete50 entrega[1]=billete20 entrega[2]=billete10 entrega[3]=billete05 entrega [4]=billete01 return entrega </pre>

Ejecución de la función: fcajero.billetes(cantidad)

>>> import fcajero >>> fcajero.billete(77) array([1, 1, 0, 1, 2]) >>>	>>> import fcajero >>> fcajero.billete(187) array([3, 1, 1, 1, 2]) >>>
--	---