

### Mejoramiento I Término 2005-2006. Septiembre 13, 2005

**Tema 4** (15 puntos) En el juego del Pozo Millonario, similar al juego de bingo, una tabla se genera con 14 números aleatorios entre 1 y 25.

Realice un algoritmo para generar una tabla de Pozo Millonario considerando que:

- Los números en la tabla NO son repetidos (10 puntos).
- Los números se presentan ordenados ascendentemente (5 puntos).

Rubrica: - Números NO repetidos (10 puntos). Presentados ordenados en forma ascendente (5 puntos).



imagen:  
<http://www.loteria.com.ec>

#### Propuesta de Solución:

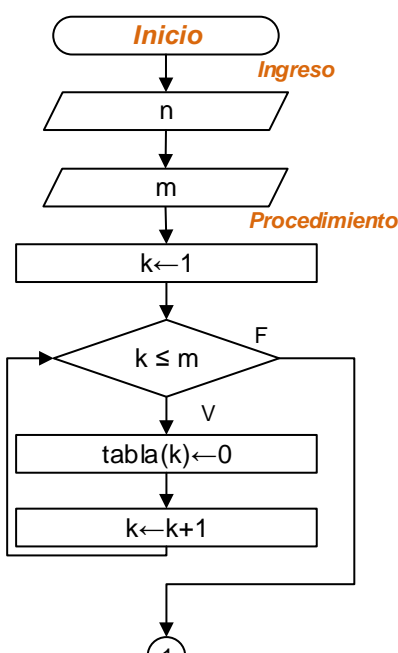
Las variables  $n$  y  $m$  se usan para ingresar los  $n$  números en la tabla entre  $m$  posibles números del ánfora. Si han observado un sorteo con un ánfora, podemos suponer que existe un panel de números con foquitos iluminados o apagados que indican si han salido del ánfora. Para simular lo mismo, se emplea un arreglo de "banderas" tipo booleana (1 o 0) que indica si el número ha salido en el sorteo.

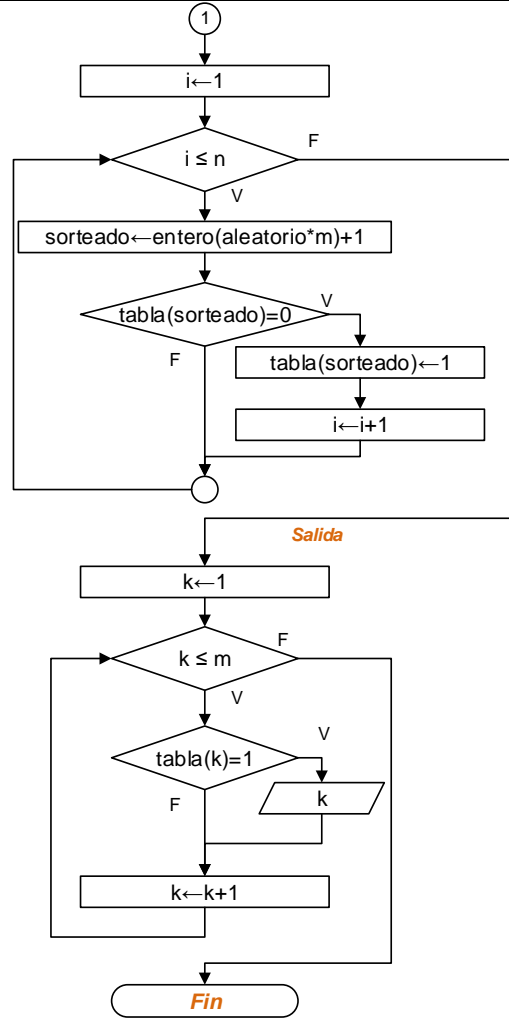
Desde luego, se inicia el tablero con valores equivalentes a ningún número seleccionado (0).

Luego de generar cada número aleatorio se marcará el número correspondiente en el tablero con 1, siempre y cuando el número no ha salido antes, repitiendo el proceso hasta completar los números necesarios.

Al final se muestra el número solamente si tiene la bandera correspondiente en afirmativo.

**Tarea: Considerar que los valores de  $n$  son menores que  $m$**

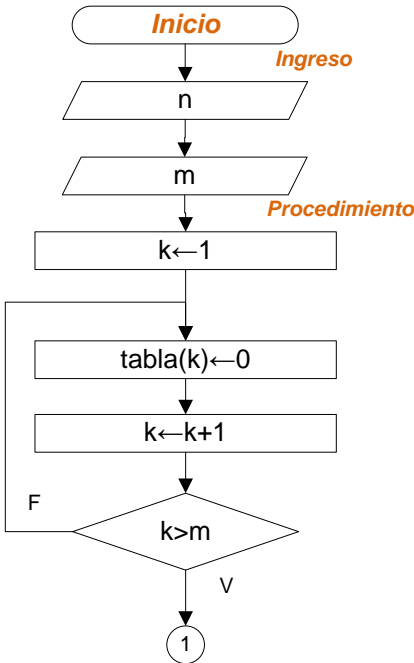
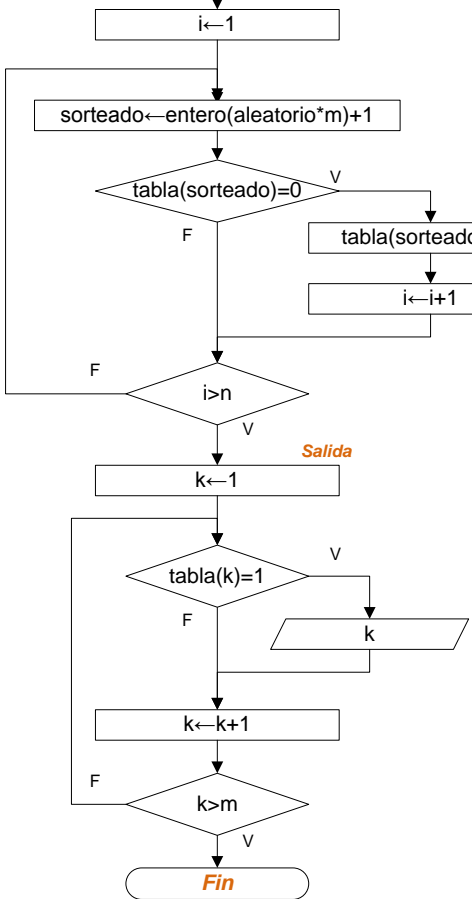
Descripción	DIAGRAMA DE FLUJO	Python
<p>Inicio</p> <p>Cantidad de <math>n</math> números por tabla</p> <p>Entre <math>m</math> disponibles (rango de selección)</p> <p>hipótesis: Ninguno seleccionado</p> <p>Primero de la tabla</p> <p>Mientras este dentro del tablero</p> <p>Marcado como libre</p> <p>Siguiente en la tabla</p> <p>Repita</p>	 <pre> graph TD     Inicio([Inicio]) -- Ingreso --&gt; n[/n/]     n --&gt; m[/m/]     m -- Procedimiento --&gt; k1[k ← 1]     k1 --&gt; k_le_m{k ≤ m}     k_le_m -- V --&gt; tabla_k[tabla(k) ← 0]     tabla_k --&gt; k_k_plus_1[k ← k + 1]     k_k_plus_1 --&gt; k_le_m     k_le_m -- F --&gt; End((1))     </pre>	<pre> # ICM00794-Fund.Computación-FCNM-ESPOL # Mejoramiento I Término 2005 # Tema 4. Pozo millonario # Propuesta: edelros@espol.edu.ec import numpy import random  n=int(input('seleccionar: ')) m=int(input('de cuantos: '))  # Procedimiento # Ninguno seleccionado tabla=numpy.zeros(m+1,dtype=int) k=1  while (k&lt;=m):      tabla[k]=0      k=k+1     </pre>

<p>Sorteo:</p> <p>Primer número</p> <p>Mientras se escoge los números</p> <p>Genera el aleatorio</p> <p>Revisa si no es repetido</p> <p>Marca número sorteado</p> <p>Siguiente sorteo</p> <p>Fin condicional repita</p> <p>SALIDA:</p> <p>Primer número</p> <p>Mientras se revisa el tablero</p> <p>Mostrar solo los seleccionados.</p> <p>Siguiente número</p> <p>Repita</p> <p>Fin de algoritmo</p>	 <pre> graph TD     Start((1)) --&gt; I1[i ← 1]     I1 --&gt; Cond1{i ≤ n}     Cond1 -- F --&gt; End1(( ))     Cond1 -- V --&gt; Gen[sorteado ← entero(aleatorio * m) + 1]     Gen --&gt; Cond2{tabla[sorteado] = 0}     Cond2 -- V --&gt; Mark[tabla[sorteado] ← 1]     Mark --&gt; IncI[i ← i + 1]     IncI --&gt; Cond1     Cond2 -- F --&gt; End1     End1 -- Salida --&gt; K1[k ← 1]     K1 --&gt; Cond3{k ≤ m}     Cond3 -- F --&gt; End2(( ))     Cond3 -- V --&gt; Cond4{tabla[k] = 1}     Cond4 -- V --&gt; PrintK[/k/]     PrintK --&gt; IncK[k ← k + 1]     IncK --&gt; Cond3     End2 --&gt; End3([Fin])                     </pre>	<pre> i=1 while (i&lt;=n):     sorteado=int(random.random()*m)+1     if (tabla[sorteado]==0):         tabla[sorteado]=1         i=i+1  #Salida k=1 print('Los numeros de la tabla son:') while (k&lt;=m):     if (tabla[k]==1):         print(k)     k=k+1                     </pre>
---	---	---

**Ejecución del algoritmo: pozomillonario.py**

<pre> &gt;&gt;&gt; seleccionar:14 de cuantos:25 Los números de la tabla son: 1 3 4 7 11 13 14 16 17 20 21 23 24 25                     </pre>	<pre> &gt;&gt;&gt; seleccionar:14 de cuantos:25 Los números de la tabla son: 1 2 3 5 7 8 10 11 17 18 19 21 22 24                     </pre>
---	---

Se presenta otra forma de resolver el ejercicio con lazos repita-hasta.

Descripción	DIAGRAMA DE FLUJO	Python
<p>Inicio</p> <p>Seleccionar <math>n</math> números</p> <p>Entre <math>m</math> disponibles (rango de selección)</p> <p>hipótesis: Ninguno seleccionado</p> <p>Primero de la tabla</p> <p>Repita</p> <p>    Marcado como libre</p> <p>    Siguiente en la tabla</p> <p>Hasta poner todos los números en cero</p>	 <pre> graph TD     Inicio([Inicio]) --&gt; Ingreso[/Ingreso/]     Ingreso --&gt; n[/n/]     n --&gt; m[/m/]     m --&gt; Proc[Procedimiento]     Proc --&gt; k1[k ← 1]     k1 --&gt; LoopStart(( ))     LoopStart --&gt; tabla0[tabla(k) ← 0]     tabla0 --&gt; kinc[k ← k + 1]     kinc --&gt; Cond{k &gt; m}     Cond -- F --&gt; LoopStart     Cond -- V --&gt; Conn1((1))     </pre>	<pre> # ICM00794-Fund.Computación - FCNM-ESPOL # Mejoramiento I Término 2005 # Tema 4. Pozo millonario # Propuesta: edelros@espol.edu.ec import numpy import random n=int(input('seleccionar: ')) m=int(input('de cuantos: '))  # Ninguno seleccionado tabla=numpy.zeros(m+1,dtype=int) k=1 while not(k&gt;m):     tabla[k]=0     k=k+1     </pre>
<p>Sorteo: Primer número</p> <p>Repita</p> <p>    Genera el aleatorio</p> <p>    Revisa si no es repetido</p> <p>    Marca número sorteado</p> <p>    Siguiente sorteo</p> <p>    Fin condicional</p> <p>Hasta completar todos los <math>n</math> números</p> <p>Primer número</p> <p>Repetir</p> <p>    Mostrar solo los seleccionados.</p> <p>    Siguiente número</p> <p>Hasta leer toda la tabla</p> <p>Fin de algoritmo</p>	 <pre> graph TD     Conn1((1)) --&gt; i1[i ← 1]     i1 --&gt; LoopStart(( ))     LoopStart --&gt; sorteado[sorteado ← entero(aleatorio * m) + 1]     sorteado --&gt; Cond1{tabla[sorteado] = 0}     Cond1 -- V --&gt; tabla1[tabla[sorteado] ← 1]     tabla1 --&gt; iinc[i ← i + 1]     iinc --&gt; LoopStart     Cond1 -- F --&gt; Cond2{i &gt; n}     Cond2 -- F --&gt; LoopStart     Cond2 -- V --&gt; Salida[Salida]     Salida --&gt; k1[k ← 1]     k1 --&gt; LoopStart2(( ))     LoopStart2 --&gt; Cond3{tabla(k) = 1}     Cond3 -- V --&gt; kout[/k/]     kout --&gt; kinc[k ← k + 1]     kinc --&gt; LoopStart2     Cond3 -- F --&gt; LoopStart2     LoopStart2 --&gt; Cond4{k &gt; m}     Cond4 -- F --&gt; LoopStart2     Cond4 -- V --&gt; Fin([Fin])     </pre>	<pre> i=1 while not(i&gt;n):     sorteado=int(random.random()*m)+1     if (tabla[sorteado]==0):         tabla[sorteado]=1         i=i+1         i=i+1  #Salida k=1 print("Los numeros de la tabla son:") while not(k&gt;m):     if (tabla[k]==1):         print(k)     k=k+1     </pre>