

Parcial II Término 2005-2006. Diciembre 06, 2005

Tema 2 (25 puntos) Considere el desarrollo del siguiente negocio:

- Una persona entrega x dólares, y se le paga 10% mensual del valor inicial depositado en forma permanente.
- Suponga que las personas **NO retiran** el dinero depositado, solo los intereses que se generan.
- La persona que recibe el dinero de los participantes usa el 20% del dinero x depositado de cada persona como comisión por gestión y gastos, quedando como saldo lo que había menos intereses y menos comisiones.
- Suponga que cada mes se duplica la cantidad de personas que invierte la misma cantidad x de dinero, con las mismas condiciones.
- Pero únicamente hay n personas que pueden entrar al negocio.

Describa un algoritmo para determinar en qué **mes** no habrá suficiente dinero para pagar a los depositantes.

Rubrica: manejo de saldos (10 puntos), control de participantes (5 puntos). Control de meses (5 puntos). Algoritmo integral (5 puntos)

Propuesta de Solución:

Para la solución, considere las operaciones descritas en el orden lógico de ejecución: depósitos, comisiones e intereses para calcular el saldo. El control de participantes e invitados permitirá calcular apropiadamente los depósitos de los invitados y los intereses pagados a los participantes.

Los meses se cuentan para cada iteración hasta que se cumpla la condición que los intereses pagados sean mayores al saldo.

Descripción	DIAGRAMA DE FLUJO	OCTAVE / MATLAB
Inicio		% Parcial II Termino 2005 % Tema 2. Problema piramidal
Cantidad por depositante		<code>x=input('cantidad por depositante:');</code>
Máximo de participantes		<code>n=input('personas en el mercado:');</code>
Contador de mes		<code>mes=0;</code>
Invitados		<code>invitado=1;</code>
Participantes		<code>participa=0;</code>
Saldo de cuentas		<code>saldo=0;</code>
Interés pagado mes		<code>interesepaga=0;</code>

<p>Mientras pueda pagar intereses</p> <p>Calcula depósitos</p> <p>Calcula comisiones</p> <p>Calcula intereses pagados a participantes</p> <p>Saldo de caja</p> <p>Participantes para el mes siguiente</p> <p>Determina invitados para el mes siguiente siempre que haya participantes</p> <p>Cuenta un mes</p> <p>Repita</p> <p>Mostrar resultado</p>	<p>1</p> <p><i>Mientras-Repita</i></p> <pre> graph TD Start((1)) --> Cond1{interespaga <= saldo} Cond1 -- V --> Proc1[deposito ← invitado * x] Proc1 --> Proc2[comision ← 0.20 * deposito] Proc2 --> Proc3[interespaga ← 0.10 * x * participa] Proc3 --> Proc4["saldo ← saldo + deposito - comision - interespaga"] Proc4 --> Proc5[participa ← participa + invitado] Proc5 --> Cond2{participa <= n/2} Cond2 -- V --> Proc6[invitado ← participa] Cond2 -- F --> Proc7[invitado ← 0] Proc6 --> Proc8[mes ← mes + 1] Proc7 --> Proc8 Proc8 --> Cond1 </pre> <p><i>Salida</i></p> <p>mes</p> <p><i>Fin</i></p>	<pre> while (interpaga <= saldo) deposito = invitado * x; comision = 0.20 * deposito; interespaga = 0.10 * x * participa; saldo = saldo + deposito - comision - interespaga; participa = participa + invitado; if participa <= (n/2) invitado = participa; else invitado = 0; end mes = mes + 1; end disp('meses con saldo disponible:'); disp(mes); </pre>
---	---	---

Ejecución del algoritmo: probpiramidal.m

<pre> >> probpiramidal cantidad por depositante:100 personas en el mercado:100 meses con saldo disponible: 14 </pre>	<pre> >> probpiramidal cantidad por depositante:100 personas en el mercado:250 meses con saldo disponible: 15 </pre>
--	--

Otra solución con Repita-Hasta

