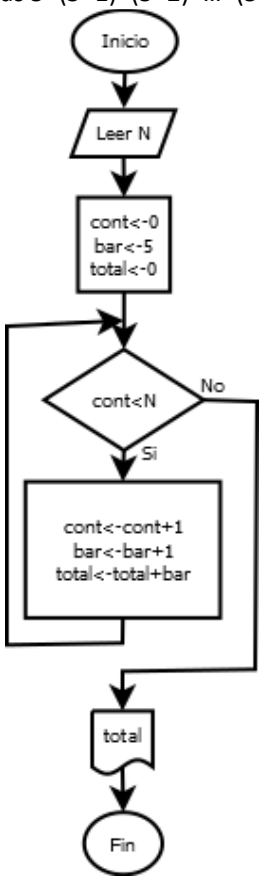
CI-2125 Computación I

Práctica 4

Contenido: Estructuras de control iterativas (while, do-while, for). Sentencias break y continue.

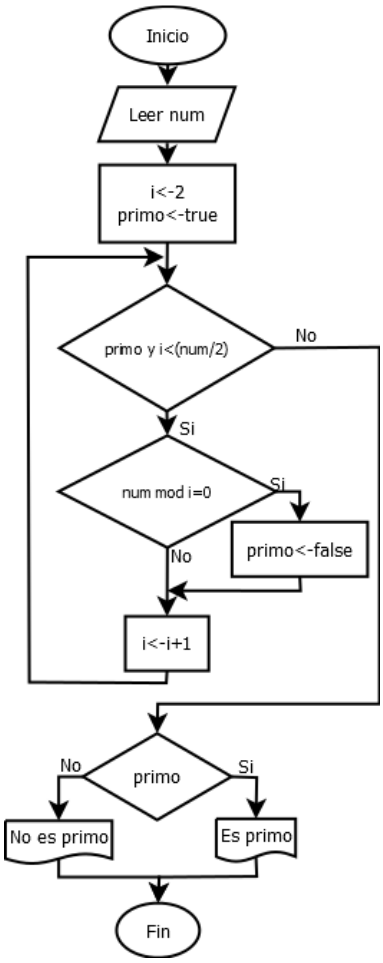
1. Suponga que un estudiante cambia 5 barajitas en un día. Cada día siguiente cambia una barajita más que el día anterior. Diseñe el diagrama de flujo de un algoritmo que determina cuántas barajitas habrá cambiado en N días. Escriba el programa equivalente en C.

   SOLUCIÓN:
   Entradas: N (entero)
   // PRE: N>0
   Salidas: total (entero)
   // POST: total es la suma de barajitas 5+(5+1)+(5+2)+…+(5+N-1)



2. Diseñe el diagrama de flujo de un algoritmo que permita determinar si un número entero positivo es primo. Traduzca el algoritmo a un programa en C utilizando un while, luego un for y finalmente un do-while.

   SOLUCIÓN:
   Entradas: num (entero)
   // PRE: num> 0
   Salidas: primo (lógico)
   // POST: primo dice si num es un número primo o no



Note que en este algoritmo, luego de la asignación primo<-false del condicional (num mod i ≠ 0), no es necesario realizar la instrucción i<-i+1, sino que directamente debería ir al chequeo de la condición de salida del ciclo. Por esto, en la traducción al lenguaje C, se puede utilizar una instrucción continue. El programa en C equivalente que utiliza la instrucción continue se llama primocontinue.c

3. Diseñe el diagrama de flujo de un algoritmo que determine el Máximo Común Divisor entre dos números enteros. El Máximo Común Divisor puede ser calculado mediante la siguiente definición conocida como el algoritmo de Euclides:

   MCD(a,a)=a

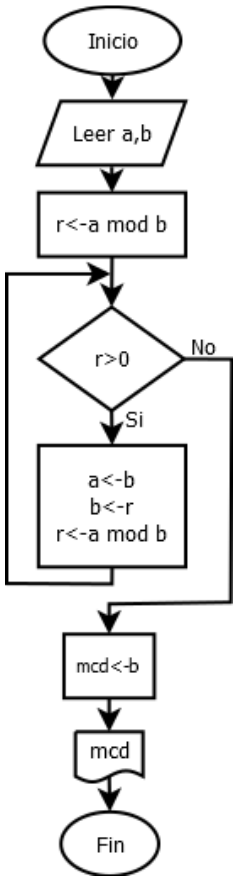
   MCD(a,b)=MCD(b,a)

   Si a = bq + r, entonces MCD(a, b) = MCD(q, r) donde q es el cociente de dividir a entre b y r es el resto.

   Note que el primer caso dice que cuando el resto es cero encontramos el máximo común divisor.

   Escriba el programa en C equivalente.

   SOLUCIÓN:
   Entradas: a,b (entero)
   // PRE: a > 0 y b > 0 y a>=b
   Salidas: mdc (entero)
   // POST: mdc tiene el valor del Máximo Común Divisor entre a y b



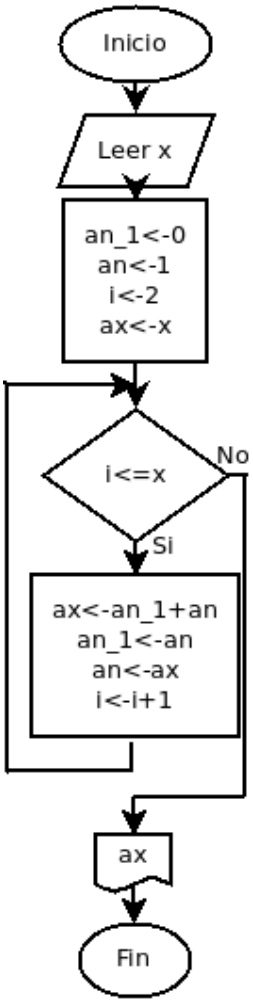
Note que en este algoritmo la instrucción r<-a mod b aparece dos veces (antes del ciclo y como última instrucción del bloque interno del ciclo). Una posible modificación al diagrama de flujo es comenzar el ciclo desde esa instrucción, luego colocar la condición de salida del ciclo luego de esa instrucción y terminar el ciclo con las dos asignaciones de a y b. Esto se traduce a lenguaje C como un ciclo sin condición o con condición vacía, y la salida del ciclo se logra con una instrucción break precedida de la condición de salida del ciclo (en este caso es r<=0). El programa en C equivalente que utiliza la instrucción break se llama mcdbreak.c

4. La serie de Fibonacci se define de la siguiente manera $a_n = a_{n-1} + a_{n-2}$. Con $a_0=0$ y $a_1=1$. Diseñe el diagrama de flujo de un algoritmo que determine el término a_x de la serie. Escriba el programa equivalente en C.

   SOLUCIÓN:

   Entradas: x (entero)
   // PRE: x >= 0

   Salidas: a_x (entero)
   // POST: a_x tiene el valor del x-ésimo término de la serie de Fibonacci



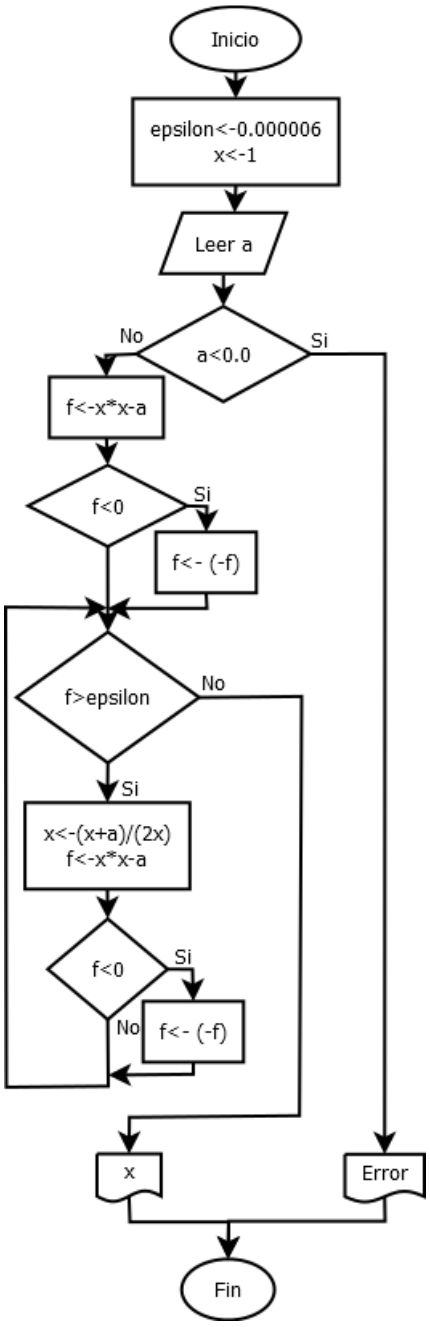
5. El método iterativo de Newton Raphson es muy útil para determinar las raíces de funciones continuas, la formula general es:

$$X_{I+1} = X_i - \frac{f(X_i)}{f'(X_i)}$$

Un método iterativo calcula el valor de X iterativamente hasta que dicho valor sea menor que un épsilon, donde épsilon representa la precisión del valor. Entre más pequeño sea el épsilon más exacta es la respuesta. Se puede tomar por ejemplo el valor inicial de épsilon=0,000006. Diseñe el diagrama de flujo de un algoritmo que determine la raíz cuadrada de un número real a utilizando el método de Newton Raphson. Para este caso la función es f(x) = $x^2 - a$ pues de esta función se deduce que sus raíces son $x = \pm\sqrt{a}$.
SOLUCIÓN:

Entradas: a (real)
// PRE: a >= 0

Salidas: x (real)
// POST: x contiene la aproximación de la raíz cuadrada de a



En esta solución pasa lo mismo que con el ejercicio 3: las instrucciones f<x*x-a y el if f<0 aparecen dos veces (antes del ciclo y como última instrucción del bloque interno del ciclo). Al igual que el ejercicio 3 se puede usar un ciclo con condición vacía que tenga un condicional para la salida del ciclo con un break. El programa en C equivalente que utiliza la instrucción break se llama NewtonRapsonBreak.c