

Examen Final

Curso: Introducción a la Ciencia de la Computación

Ciclo: 2016.1

1. [4 ptos.] Sobre el siguiente programa:

```
1  #include <stdio.h>
2
3  int main ( )
4  {
5      if (-1)
6          printf("menos uno.\n");
7      else
8          if (2.2)
9              printf("dos punto dos.\n");
10     else
11         printf("tres...\n");
12     return 3;
13 }
```

- (a) ¿Cuántas sentencias if hay? ¿Por qué?
- (b) ¿Qué es lo que se mostraría en la pantalla de la computadora al ejecutarlo?
- (c) Si en la línea 5 se cambia el -1 por el 0, ¿qué se mostraría en la pantalla?
- (d) Si en las líneas 5 y 8 se cambiaran dichos números por el 0, ¿qué se mostraría en la pantalla?
2. [5 ptos.] Implementar un programa en C que solicite la cota inferior, la cota superior y un número que entran como datos, evalúa los datos y muestra uno de los siguientes mensajes
- Está dentro de las cotas.
 - Supera la cota superior.
 - Supera la cota inferior.

Por ejemplo, si ingresa como cota inferior, cota superior y número los valores de 1, 3 y 3, respectivamente, entonces el programa le debe mostrar: Está dentro de las cotas.

3. [5 ptos.] El **método de bisección** es un algoritmo de búsqueda de raíces que trabaja dividiendo el intervalo a la mitad y seleccionando el subintervalo que tiene la raíz. Se basa en el teorema del valor intermedio, el cual establece que toda función continua f en un intervalo cerrado $[a, b]$, con $a < b$, toma todos los valores que se hallan entre $f(a)$ y $f(b)$. Esto es que todo valor entre $f(a)$ y $f(b)$ es la imagen de al menos un valor en el

intervalo $[a, b]$. En caso de que $f(a)$ y $f(b)$ tengan signos opuestos, el valor cero sería un valor intermedio entre $f(a)$ y $f(b)$, por lo que con certeza existe un p en $[a, b]$ que cumple $f(p) = 0$. De esta forma, se asegura la existencia de al menos una solución de la ecuación $f(x) = 0$. A continuación, presentamos un pseudocódigo sobre el método de bisección para la función $f(x) = \cos(x) - x$:

```

1  Inicio
2      leer (a)
3      leer (b)
4      leer (error)
5      f(a) = cos(a) - a
6      f(b) = cos(b) - b
7      si (f(a)*f(b) > 0) entonces
8          imprimir ("No se puede aplicar el metodo")
9      sino
10         iter = techo( log_2( (b-a)/error ) )
11         para (i = 1) hasta (iter) hacer
12             x = (a+b) / 2
13             f(x) = cos(x) - x
14             f(a) = cos(a) - a
15             si (f(a)*f(x) < 0) entonces
16                 b = x
17             sino
18                 a = x
19             fin si
20         fin para
21         imprimir ("La raz es: ")
22         imprimir (x)
23     fin si
24 Fin

```

- (a) El algoritmo funciona siempre que no se encuentre una raíz exacta al momento de dividir el intervalo. ¿Qué cambios haría para que el algoritmo funcione con raíces exactas?
- (b) El lenguaje C no admite logaritmo en base 2. ¿Cómo implementaría la sentencia o instrucción de la línea 10?

Sábado 9 de Julio.