

# Lenguajes de Programación

Dr. Carlos Artemio Coello Coello  
Tarea No. 5  
3 de marzo de 2016

## Programación en Scheme

1. (**20 puntos**) Defina un procedimiento **acronimo** que tome como argumento una cadena **strng** y regrese el acrónimo correspondiente al título dado. Para hacer las cosas más interesantes, su procedimiento debiera de ser capaz de deshacerse de las siguientes proposiciones (en inglés): **at**, **the**, **in**, **of**, **and**, **for** y **with**. Sólo se probará el procedimiento con frases en inglés. Pruebe su implementación con los siguientes ejemplos:

(**acronimo** "The United States of America")  $\implies$  "USA"  
(**acronimo** "Association for Computing Machinery")  $\implies$  "ACM"  
(**acronimo** "Structure and Interpretation of Computer Programs")  
 $\implies$  "SICP"  
(**acronimo** "Tools with Artificial Intelligence")  $\implies$  "TAI"  
(**acronimo** "University of Alabama in Huntsville")  $\implies$  "UAH"  
(**acronimo** "University of California at Los Angeles")  $\implies$  "UCLA"  
(**acronimo** "Quod Erat Demonstrandum")  $\implies$  "QED"

2. Un tal reverendo Zeller desarrolló una fórmula para calcular el día de la semana para cualquier fecha del calendario Gregoriano. Las entradas del algoritmo se deben especificar de la siguiente manera:

- $m$  es el mes del año, de tal manera que marzo se representa como  $m = 1$ . Enero y febrero son los meses 11 y 12 del año anterior.
- $d$  es el día del mes.
- $y$  es el año del siglo.
- $c$  es el siglo anterior.

Por ejemplo, para representar el 4 de julio de 1989, usaríamos  $m = 5$ ,  $d = 4$ ,  $y = 89$  y  $c = 19$ . Por su parte, para representar 25 de enero de 1989, usaríamos  $m = 11$ ,  $d = 25$ ,  $y = 88$  y  $c = 19$ . El algoritmo para calcular el día de la semana es el siguiente:

- (a) Tomar la parte entera de  $(13m - 1)/5$ . Llamar a esto  $a$ .

- (b) Tomar la parte entera de  $y/4$ . Llamar a esto  $b$ .
- (c) Tomar la parte entera de  $c/4$ . Llamar a esto  $e$ .
- (d) Calcular  $f = a + b + e + d + y - 2c$ .
- (e) Hacer  $r$  igual a  $f$  modulo 7.
- (f)  $r$  nos dice el día de la semana, de manera que el domingo corresponde a  $r = 0$ , el lunes a  $r = 1$ , etc.

(30 puntos) Escriba un programa en Scheme llamado **dia-semana**, que pida el mes, el día y el año. El mes debe introducirse de la manera habitual (p.ej., enero sería el mes 1 y diciembre sería el mes 12). El año también debe introducirse de la manera habitual (p.ej., 1998). El programa deberá entonces convertir estos datos al formato requerido por el algoritmo y calcular el día de la semana. El programa deberá también preguntar al usuario si desea dar otra fecha como entrada y deberá terminar cuando el usuario responda **no**. La salida de su programa debe lucir exactamente como se indica a continuación (se muestra en negritas lo que el programa imprime y en letras normales los datos que el usuario proporciona):

(**dia-semana**)

**Introduzca el mes (1..12):** 2

**Introduzca el día (1..31):** 24

**Introduzca el a#o:** 2005

**El 24/2/2005 es un** Jueves

**Desea proporcionar otra fecha?** si

**Introduzca el mes (1..12):** 5

**Introduzca el día (1..31):** 10

**Introduzca el a#o:** 1980

**El 10/5/1980 es un** Sabado

**Desea proporcionar otra fecha?** si

**Introduzca el mes (1..12):** 1

**Introduzca el día (1..31):** 13

**Introduzca el a#o:** 1989

**El 13/1/1989 es un** Viernes

**Desea proporcionar otra fecha?** si

**Introduzca el mes (1..12):** 10

**Introduzca el día (1..31):** 24

**Introduzca el a#o:** 2000

**El 24/10/2000 es un** Martes

**Desea proporcionar otra fecha?** no

**Eso es todo, amigos!!!**

Es importante aclarar que el algoritmo proporcionará respuestas aún para fechas no válidas (p.ej., mes 24). Para fines de la tarea, no es necesario interceptar estos errores.

3. (**Bonificación: 20 puntos**) Escriba un procedimiento INTERACTIVO llamado **numero-perfecto**, que tome como entrada un entero y determine si es perfecto o no. Un número es *perfecto* si la adición de todos sus factores, excluyendo el número mismo, genera dicho número. La salida de su programa debe lucir exactamente como se indica a continuación (se muestra en negritas lo que el programa imprime y en letras normales los datos que el usuario proporciona):

```
(numero-perfecto)
Deme un numero: 10
Los factores de 10 son: 1 2 5
El numero 10 no es perfecto.
Quiere proporcionar otro numero? si
```

```
Deme un numero: 6
Los factores de 6 son: 1 2 3
El numero 6 es perfecto.
Quiere proporcionar otro numero? si
```

```
Deme un numero: 28
Los factores de 28 son: 1 2 4 7 14
El numero 28 es perfecto.
Quiere proporcionar otro numero? si
```

```
Deme un numero: 496
Los factores de 496 son: 1 2 4 8 16 31 62 124 248
El numero 496 es perfecto.
Quiere proporcionar otro numero? no
```

**Hasta la vista!!!**

Le resultará útil leer el capítulo 6 de los apuntes de Scheme que están disponibles en la página web del curso.

La solución a TODOS estos problemas deberá incluirse en un solo archivo ASCII (llamado **sol-tarea5.scm**) en un diskette que se entregará junto con el resto de su tarea. Asegúrese de incluir también una impresión del código fuente de sus procedimientos dentro de su tarea. Las soluciones deben implementarse con los nombres que se indican en cada problema de los antes enumerados.

**Fecha de entrega:** *Martes 15 de marzo a las 12:00hrs.* Toda tarea entregada tarde será penalizada con 10% (sobre la calificación obtenida) por cada periodo de 24 horas que se retrase su entrega.