

Tarea de LISP: Calculadora Histórica

PROF. KIRSTEIN GÄTJENS S.

“La diferencia entre el poeta y el matemático es que el poeta intenta meter su cabeza en los cielos, mientras que el matemático intenta meter los cielos en su cabeza”

– G.K. Chesterton

Instrucciones generales:

- Entregue un archivo de texto con el fuente del programa solicitado. Use extensión .scm.
- Debe entregarse antes del miércoles 13 de marzo a la medianoche al correo kirstein.evaluaciones@gmail.com
- El nombre del scm debe ser consistente con los estándares de clases:
Calculadora-Histórica-Apellido-Nombre.scm
- El subject del correo es lo usual: TAREA: Calculadora Histórica
- El cuerpo del correo debe contener, como todo correo del curso, su nombre completo, número de carné y curso.
- La documentación a entregar debe ser en el mismo fuente entre comentarios. Tiene 3 partes mínimas, pero si requiere de más explicaciones son bienvenidas. Usualmente ayudan al asistente a poner una nota mejor.
- Debe tener una pequeña portada debe estar al inicio del mismo fuente entre comentarios. Un manual de usuario que explique como se puede probar su tarea con detalle (cuáles son las funciones de entrada y parámetros) y un análisis de resultados con el sistema de ABCDE usado en el curso.
- Al ser una tarea de un lenguaje funcional es indispensable que el código de cada función posea oportunos comentarios. Precondiciones o poscondiciones necesarias, parámetros y breve descripción.
- Esta tarea pequeña vale por 9 resúmenes.
- Es una tarea mediana, pues el objetivo que tiene es que practiquen la programación funcional de una forma inicial, pensamos dejar una tarea más grande en el otro lenguaje funcional.
- Deben trabajar en chicken.
- Al ser una tarea del paradigma funcional es absolutamente restringido el uso de elementos imperativos, tales como variables, iteraciones o secuencias.

La tarea consiste en realizar un programa que sea una calculadora que trabaje con varias numeraciones históricas y con una de las numeraciones del curso de FOC.

Las numeraciones a trabajar de forma obligatoria son: Números Romanos, Egipcios, Mayas, Babilónicos y Chinos y Kulkán (las dos versiones).

Además se debe trabajar con lo que llamaremos un número tradicional. Un número tradicional está escrito en el sistema posicional pero con la base indicada en decimal al inicio.

Acá explicaremos la nomenclatura para representar las numeraciones históricas solicitadas. Usted debe pensar la nomenclatura para la numeración que se le asignó.

La calculadora debe implementar las siguientes funciones:

Suma, Resta, Multiplicación, DivisiónEntera, Residuo y Complemento. Esta última no se debe implementar para las numeraciones históricas. Estos son los nombres de las funciones de entrada esperadas.

Se espera que las funciones de entrada sean las operaciones. Deben ser operaciones al estilo de LISP, es decir análogas a las funciones +, -, *, / o reminder.

Al invocar una función se debe indicar la numeración de la respuesta seguida de los operandos solicitados.

El programa deberá ser capaz de reportar un error en la representación de los números y nunca generar un número de respuesta que sea inconsistente con la representación. Si se detecta un error se debe retornar como dato un string con el mensaje significativo de error.

Las diferentes numeraciones deben trabajar solo con los números de su posible rango de representación. Es posible entonces que el resultado de una operación sea "overflow".

Para representar los diferentes números tendremos siempre pares ordenados, donde el primer elemento es el tipo de numeración o base y el segundo los datos del número. Los detalles de cada uno son:

Números tradicionales: Es una lista con dos elementos donde el primer es la base en decimal (entre 2 y 36) y el segundo es un string con el número usando la nomenclatura tradicional de letras.

(16 "4CBF1")

Números Romanos: Es un par ordenado donde el primer elemento es el símbolo R. El segundo elemento es una lista con la secuencia de símbolos que representan el número. Los romanos no usaban el cero.

(R (X V I J I))

Números Egipcios: Como los números egipcios no tienen orden solamente se debe sumar los dígitos representados. Al generarlo deben lograr de que sea un número razonablemente simétrico para resaltar la belleza buscada con los egipcios. Usaremos la E para indicar que es un número egipcio y las letras siguientes en orden para los dígitos: B, A, C, F, D, E y R para el Baston, Asa, Cuerda, Flor de Loto, Dedo, Esclava y Ra. El cero es la lista vacía.

(E (R R C C A B A C C R R)) representa el número 4000421

Números Mayas: Usaremos listas para los dígitos mayas. Donde los elementos básicos son * y el | (asterisco y pipe). La lista vacía será el cero. (| | | *) sería el dígito 16. La M será el símbolo que indica que es un número maya. Hay que especificar que cambiamos los | por ! debido a errores con el pipe al ejecutar el programa

(M (() (| | * * *) (| *))) es el número 2660 (0x1+13x20+6x400)

Números Babilónicos: Usaremos la B como símbolo. Como solo trabajan con dos cuñas usaremos < y ^.

(B ((< ^) () (^ ^ ^))) Representa al número 39603 (3x1+0x60+11x3600)

Números Chinos: Utilizaremos la C para indicar que es un número chino y usaremos los nombres de los números como símbolos para armar el número completo. Ignoren tildes y apóstrofes.

(C ((wan qi)(quian san)(ba))) es el número 70308

Números en KuKulKán: Representaremos la K como símbolo de la numeración. La espiral la mantendremos como una lista de tres listas de tres elementos. Cada dígito de la numeración será su nombre en maya.

(K ((chicchan manic ix) (cib lamat muluc) (akbal oc imix))) representa al número 8570002099 (es el primer ejemplo en el libro de FOC).

Para manejar las espirales de espirales usaremos al símbolo KG (Kukulkán Grande). Se tendrá la misma lista con tres elementos para cada fila o sea una lista de tres listas de tres elementos, solo que ahora cada dígito no será una palabra sino un número de la numeración Kukulkán básica. Por ejemplo:

(KG (((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix))) (((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix))) (((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix)) ((chicchan manic ix) (cib lamat muluc) (akbal oc imix))) sería el número formado por los 9 dígitos usando el número anterior.

Para la numeración que se le asignó debe diseñar una representación para el número que calce con las acá descritas.

Ejemplos de invocaciones a las funciones de la tarea:

```
>(suma R (2 "1010") (B ((<<<^^)(<^^^^^^^^^^))))  
(R (M C M X L I X))
```

Para obtener el complemento haga:

```
>(complemento 2 (10 "907"))  
(2 "01011101")
```

1- Reste el dígito mayor de la base ingresada con cada dígito del número ingresado

Ejemplo: 10 (9 0 7)

9 - 9 = 0

9 - 0 = 9

9 - 7 = 2

R/ 10 (9 2) -> 2 (0 1 0 1 1 1 0 1)

¡Suerte!
