

Relación de ejercicios nº 3: Temas 7 - 11.

1. Modifica el ejercicio 10 de la relación 2 para que el programa pregunte al usuario qué características desea que tenga la planta y éste las introduzca por teclado, y para que la salida indicando qué plantas pueden plantarse se dé también por pantalla.
2. Escribe un programa que lea un fichero de texto que contenga nombres de personas y sus respectivas edades, y que cree otro fichero que contenga los mismos datos en orden ascendente de edad. El programa debe pedir al usuario los nombres de los ficheros de entrada y de salida. Por ejemplo, un fichero que contenga

```
Felipe Sagasta 37
Ana de la Haba 35
Jacobó Quena 36
Gabriel Acero 34
```

debería dar lugar al fichero de salida

```
Gabriel Acero 34
Ana de la Haba 35
Jacobó Quena 36
Felipe Sagasta 37
```

3. Define una función llamada *subcadena* para obtener una subcadena de una cadena de caracteres. El primer argumento debe ser la cadena. Los dos siguientes argumentos deben ser los índices de la subcadena a extraer *i*, *f* (inicio y fin), teniendo en cuenta las siguientes consideraciones:
Si $i > f$, devolverá la subcadena desde 1 a *i*.
Si no se da *f*, devolverá desde *i* hasta el final de la cadena original.
4. Define un par de funciones, llamadas *cabeza* y *cola*. Las dos reciben como argumento una cadena de caracteres y un entero *n*. Devolverán los *n* primeros o últimos caracteres de la cadena, respectivamente.
5. Escribe una función (*es-potencia* <número> <base>) que devuelva TRUE si el primer argumento es una potencia entera del segundo y FALSE en caso contrario. Tanto los argumentos como el exponente deben ser enteros ≥ 0 . **NOTA:** Cuidado cuando alguno de los dos argumentos valga 1.
6. Haz un programa que use la función definida en el ejercicio anterior para, dada una serie de hechos (*numero* <número>), haga la suma de todos los números que son potencia del más pequeño (incluido él mismo), mostrando por pantalla la lista de valores sumados y el resultado de la suma. Fíjate en el siguiente ejemplo:

```
CLIPS> (facts)
f-0      (initial-fact)
f-1      (numero 2)
f-2      (numero 3)
f-3      (numero 4)
f-4      (numero 5)
f-5      (numero 6)
f-6      (numero 7)
f-7      (numero 8)
f-8      (numero 9)
f-9      (numero 10)
For a total of 10 facts.
CLIPS> (run)
La suma de (8 4 2) es 14
```

7. Escribe un programa que traduzca un mensaje en código Morse a la correspondiente serie de caracteres alfabéticos. Fíjate en el ejemplo de salida del programa. El punto se representa mediante el asterisco '*' y la raya mediante el guión '-'. Los mensajes deben introducirse como cadenas de caracteres sin ningún espacio en blanco entre medias. No utilices prioridades, módulos ni funciones procedimentales (salvo, en todo caso, *bind*).

```
CLIPS> (run)

Introduzca mensaje (Enter para terminar): ***/---/***
El mensaje es: SOS
Introduzca mensaje (Enter para terminar):
CLIPS>
```

El código Morse es:

| | |
|------|---|
| *_ | A |
| _*_* | B |
| _*_* | C |
| _* | D |
| * | E |
| **_* | F |
| --* | G |

| | |
|------|---|
| **** | H |
| ** | I |
| *--- | J |
| _*_ | K |
| *_** | L |
| -- | M |
| _* | N |

| | |
|------|---|
| --- | O |
| *_** | P |
| --*_ | Q |
| *_* | R |
| *** | S |
| - | T |
| **_ | U |

| | |
|------|---|
| ***_ | V |
| *_-- | W |
| _*** | X |
| _*- | Y |
| --** | Z |

8. Repite el programa del ejercicio anterior pero utilizando módulos.