Comenzado el	Friday, 19 de May de 2023, 16:10
Estado	Finalizado
Finalizado en	Friday, 19 de May de 2023, 17:58
Tiempo empleado	1 hora 48 minutos
Puntos	102,65/200,00
Calificación	51,33 de 100,00

Pregunta 1

Finalizado

Se puntúa 40,00 sobre 100,00

Se conocen los alimentos con su valor calórico, el instituto argentino de nutrición (IAN) desea conocer cuales serían las dietas posibles para un día de una persona que padece diabetes de tipo 1 y puede consumir hasta 500 calorías en total, una dieta para carnívoros y otra para vegetarianos. La lista de los alimentos se debe mostrar de menor calorías a mayor. Por lo tanto, el procedimiento recibe una lista de listas, cada lista contiene el alimento, el grupo al que pertenece y su valor calórico y debe retornar dos listas con los alimentos ordenados de menor a mayor, según su valor calórico y no superar las 500 Calorías. una lista para carnívoros y otra para vegetarianos, teniendo en cuenta que los carnívoros no consumen vegetales y los vegetarianos no consumen carne, aunque hay alimentos que consumen "ambos" y se consideran para carnívoros como para vegetarianos.

EJ.:
PROCEDIMIENTO
(dieta '(("papa" "vegetariano" 80) ("milanesa de carne" "carnívoro" 150) ("jugo de naranja" "vegetariano" 70) ("calabaza" "vegetariano" 90) ("costeleta" "carnívoro" 180) ("tiramisú" "ambos" 150) ("pastas" "ambos" 100) ("gaseosa" "ambos" 220) ("pan" "ambos" 110) ("chocolate" "ambos" 400) ("espárragos" "vegetariano" 60) ("empanadas de carne" "carnívoro" 200))

RESULTADO
(("espárragos" "jugo de naranja" "papa" "calabaza" "pastas") ("pastas" "pan" "milanesas de carne"))

```
(define esDelTipo(lambda (l clave)
          (equal? (car (cdr l)) clave)))

(define concatenar ( lambda (l1 l2)
            (if (empty? l1 )
              l2
              (cons (car l1) (concatenar (cdr l1) l2))
              )))

(define obtenerTipoLista(lambda(l clave)
              (if (empty? l)
                '()
                (if (esDelTipo (car l) clave)
                  (cons (car l) (obtenerTipoLista (cdr l) clave))
                  (obtenerTipoLista (cdr l) clave)
                )
               )
              )
 )

;TEST | ANDA
;(obtenerTipoLista '(("papa" "vegetariano" 80) ("milanesa de carne" "carnívoro" 150) ("jugo de naranja" "vegetariano" 70) ("calabaza" "vegetariano" 90) ("costeleta" "carnívoro" 180) ("tiramisú" "ambos" 150) ("pastas" "ambos" 100) ("gaseosa" "ambos" 220) ("pan" "ambos" 110) ("chocolate" "ambos" 400) ("espárragos" "vegetariano" 60) ("empanadas de carne" "carnívoro" 200)) "vegetariano")

(define obtenerPeso (lambda (elem)
        (car (cdr (cdr elem)))))

(define obtenerElemMasChico(lambda (lista compara)
                (if (empty? lista)
                  compara
                  (if (> (obtenerPeso compara) (obtenerPeso (car lista)))
                    (obtenerElemMasChico (cdr lista) (car lista))
                    (obtenerElemMasChico (cdr lista) compara) ))))
;(obtenerElemMasGrande '(("papa" "vegetariano" 80) ("milanesa de carne" "carnívoro" 150) ("jugo de naranja" "vegetariano" 70)) '("foo" "bar" 10))
(define eliminarElemento(lambda (l elem)
```

```
                (if (empty? l)
                  '()
                  (if (equal? (car l) elem)
                    (cdr l)
                    (cons (car l) (eliminarElemento (cdr l) elem))))))
;> (eliminarElemento '(("papa" "vegetariano" 80) ("milanesa de carne" "carnívoro" 150)) '("milanesa de carne" "carnívoro" 150))


(define ordenar-aux(lambda(l)
            (if (empty? l)
              '()
              (let ((chico (obtenerElemMasChico l (car l))))
               (let ((l2 (eliminarElemento l chico)))
                (cons chico (ordenar-aux l2)))))))


(define ordenar(lambda(x)
          (if (empty? x)
            '()
            (ordenar-aux x))))
(define dieta(lambda(l)
          (let ((listaVeg (concatenar (obtenerTipoLista l "vegetariano") (obtenerTipoLista l "ambos")))(listaCarn (concatenar
(obtenerTipoLista l "carnívoro") (obtenerTipoLista l "ambos"))))
           (list (cons (ordenar listaVeg) '()) (cons (ordenar listaCarn) '() ))
           )))
```

nahuelgareis.rkt

Comentario:
No resuelve, evaluando las funciones por separado se consideró la totalidad de la función que distingue la dieta, la función que ordena solo una parte porque está sacada del contexto del ejercicio.

Pregunta 2

Finalizado

Se puntúa 62,65 sobre 100,00

La cátedra de TecProg les solicita a sus alumnos que creen un programa en el paradigma logico llamado "passValidator" que se encargue de validar una contraseña de usuario. La salida de la función debe ser un string que indique si la clave ingresada no cumple con alguna de las condiciones, especificando cada una, o en caso contrario, notificar que la clave es válida.

Las condiciones que debe cumplir la misma son:

- La clave debe ser un string.
- La clave debe contener entre 8 y 12 caracteres.
- La clave debe contener al menos una letra mayúscula.
- La clave NO debe contener secuencias de 3 números ascendentes o descendentes.

EJEMPLOS:

passValidator(12). --> La entrada no es un String.

passValidator("pass"). --> La clave debe contener entre 8 y 12 caracteres.

passValidator("password"). --> La clave debe contener al menos una letra mayúscula.

passValidator("pass456word"). --> La clave NO debe contener secuencias de 3 números ascendentes o descendentes.

passValidator("passwordA"). --> ¡¡Clave valida!!

OJO: Solo pueden usar las funciones predefinidas string/1, number/1, string_chars/2 y char_code/2. El resto de las funciones deberán hacerlas Uds.

```
contar([],0).
contar([_|CDR],COUNT):-
  contar(CDR,COUNTR),
  COUNT is COUNTR+1.

tineNumeros([CAR|_]):- number(CAR), !.
tineNumeros([CAR|CDR]):- not(number(CAR)), number(CDR).

tieneSecuencia([CAR1,CAR1-1,CAR1-2|_]):- ! .
tieneSecuencia([CAR1,CAR1+1,CAR1+2|_]):- ! .
tieneSecuencia([_,_,_|CDR]):- tieneSecuencia(CDR) .

tieneMayus([CAR|_]):- char_code(CAR,NUM),NUM > 64 , char_code(CAR,NUM) ,NUM< 90, !.
tieneMayus([_|CDR]):- tieneMayus(CDR).

passValidator(WORD):-
  string(WORD),
  string_chars(WORD,L),
  contar(L,COUNT),
  COUNT > 8,
  COUNT < 12,
  not(tieneSecuencia(L)),
  tieneMayus(L).
```

nahuelgareis.pl

Comentario:

- En la validación de secuencia haces 2 veces la misma ligadura con char_code/2.
- La lógica de la validación de secuencia esta bien encarada pero mal evaluada, las comparaciones se hacen con comparador aritmético.
- Faltan reglas en el wrapper para salir con el mensaje correcto según el caso.
- Los comentarios y ejemplos de ejecución nuca están de mas.

◄ Va a asistir al parcial del tema PARADIGMA LOGICO

Ir a...

PARCIAL FUNCIONAL/LÓGICO TURNO 18:00hs 18/05/2023 ►