Ejercicios entrega grupal

**Ejercicio 1:**

**Algoritmo:** Sucesor día de la semana

**Pseudocódigo:**

-Estructura para el tipo de día

(lunes, martes, miércoles, jueves, viernes, sábado, domingo)

(1, 2, 3, 4, 5, 6, 7)

-Fin de la estructura de día

-Sucesor de día = Día (en la semana)

->Realización (siguiendo el valor del día hacer)

= “lunes”:

Resultado <- “martes”

=” martes”:

Resultado <- “miércoles”

=” miércoles”:

Resultado <- “jueves”

=” jueves”:

Resultado <- “viernes”

=” viernes”:

Resultado <- “sábado”

=” sábado”:

Resultado <- “domingo”

=” domingo”:

Resultado <- “lunes”

Si no

Nada

Fin de hacer

->Postcondición

Día = “lunes” => Resultado = “martes”

Día = “martes” => Resultado = “miércoles”

…

->Fin de sucesor

**Ejercicio 2:**

**Algoritmo**: Clasificar tres datos comparables (clasificar ‘a’, ‘b’ y ‘c’ en orden creciente)

**Pseudocódigo:**

**Entrada**  
 a, b, c: **T** → **COMPARABLE  
  
precondición**  
 ninguna **realización  
 si** a > b **entonces intercambiar** (a, b) **fin si** *# a ≤ b; situar `c'* **si** b > c **entonces intercambiar** (b, c) **fin si** *# a ≤ c; b ≤ c* **si** a > b **entonces intercambiar** (a, b) **fin si** *# a ≤ b ≤ c* **postcondición**  
 a ≤ b ≤ c **fin clasificar3**

...  
**variable**  
 c, d : (**T**, +, x) → **COMPARABLE**  
  
**realización**  
 ...  
 c ← a + b ; d ← a x b  
 **clasificar4**(a, b, c, d)  
 # a ≤ b ≤ c ≤ d  
...

*# a ≤ b ≤ c ; situar `d'*

**si**  
 d < a  
**entonces**  
 **intercambiar** (a, d) *# a ≤ d ; b ≤ c*  
 **clasificar3** (b, c, d) *# a ≤ b ≤ c ≤ d*  
**si no**  
 *# a ≤ d ; b ≤ c*  
 **clasificar3**(b, c , d) *# a ≤ b ≤ c ≤ d*  
**fin** **si**  
*# a ≤ b ≤ c ≤ d*

*# a ≤ b ≤ c; situar `d'*  
**si**  
 d < a  
**entonces**  
 *# a > d; b ≤ c*  
 **intercambiar** (a, d)  
 *# a ≤ d; b ≤ c*  
**fin** **si**  
*# a ≤ d; b ≤ c*  
**clasificar3**(b, c, d)  
*# a ≤ b ≤ c ≤ d*

**Algoritmo final**: Clasificar cuatro datos comparables

Algoritmo **clasificar4** *# Clasifica `a', `b' `c' y `d' en orden creciente.* **Entrada**  
 a, b, c, d: **T** → **COMPARABLE  
  
precondición**  
 VERDADERO **realización**  
 **clasificar3**(a, b, c)  
 *# a ≤ b ≤ c; situar `d'*  
  
 **si**  
 d < a  
 **entonces**  
 *# a > d; b ≤ c*  
 **intercambiar** (a, d)  
 *# a ≤ d ; b ≤ c*  
 **fin** **si**  
 *# a ≤ d ; b ≤ c*  
  
 **clasificar3**(b, c , d)  
 *# a ≤ b ≤ c ≤ d*  
  
**postcondición**  
 a ≤ b ≤ c ≤ d  
 **fin clasificar4**

...

**variables**

a, b, suma, producto: **ENTERO**

**inicializaciones**

a ← ??? ; b ← ???

**realización**

suma ← a + b

producto ← a x b

**clasificar4**(a, b, suma, producto)

...

**Ejercicio 3:**

**Algoritmo**: Importe descuento

**Entrada**:

precio (real) # precio del producto

**Resultado**: decimal

**Precondición**:

Precio ≥ 0

**Realización**:

**Si**:

precio < 100

**Entonces**: # No hay descuento

Resultado: 0,00

**Si no**:

**Si**:

precio ≤ 500

**entonces**: # precio entre 100 y 500, por tanto, descuento del 5%

**Resultado**: precio × 0,05

**Si no**: # precio mayor que 500, entonces hay un descuento del 8%

**Resultado**: precio × 0,08

**Postcondición**:

precio < 100 => Resultado: 0,00

precio ≤ 500 => Resultado: 0,05

precio > 500 => Resultado: 0,08

**Ejercicio 4:**

**Algoritmo:** Media

**Entrada:**

Nota1: int

Nota2: int

Nota3: int

Nota4: int

**Precondición:**

las notas se comprenden entre 0 y 20

**Realización:**

Media(int) = (nota1 + nota2 + nota3 + nota4)/4

Si media > 15:

mensaje(string) = “alumno con talento”

Si 12 <= media >= 15:

mensaje(string) = “con capacidad”

Else:

mensaje(string) = “debe reorientarse”

**Resultado:**

Mensaje: string #el mensaje que se imprime en pantalla dependiendo de la media

Media: int #la media de las notas

**Ejercicio 5**

**Algoritmo**: Importe de descuento según unidad familiar

**Pseudocódigo:**

Si niño = 0 -> No hay descuento

Si niño = 1 -> No hay descuento

Si niño = 2 -> Descuento 10%

Si niño = 3 -> Descuento 15%

Si niño = 4 -> Descuento 18%

Si niño > 4 -> Descuento de niño = 4 + 1% adicional por cada niño añadido.

**Ejercicio 6**

**Algoritmo**: Calculo del porcentaje de descuento

**Entrada**:

componentes (real) # Número de componentes que compran

cliente (booleano) # Tipo de cliente

**Resultado**: porcentaje

**Precondición**:

componentes > 0

cliente = COMMAQ o BEL

**Realización**:

**Si:**

componentes < 10.000

**entonces**: # no tiene descuento

**Resultado**: 0%

**Si**:

cliente = COMMAQ

**entonces**: # no tiene descuento

**Resultado**: 0%

**Si no**: # cliente es BEL

**entonces**: # se le añade un 1%

**Resultado**: 1%

**Si no**:

**Si**: # componentes están entre 10.000 y 20.000

componentes < 20.001

**entonces**: # tiene descuento

**Resultado**: 10%

**Si**:

cliente = COMMAQ

**entonces**: # se descuenta un 2% del 10%

**Resultado**: 8%

**Si no**: # cliente es BEL

**entonces**: # se le añade un 1% al 10%

**Resultado**: 11%

**Si no**:

**Si**: # cantidad entre 20.001 y 40.000

componentes < 40.000

**entonces**: # tiene un descuento

**Resultado**: 15%

**Si**:

cliente = COMMAQ

**entonces**: # se descuenta un 2% del 15%

**Resultado**: 13%

**Si no**: # cliente es BEL

**entonces**: # se le añade un 1% al 15%

**Resultado**: 17%

**Si no**: # cantidad superior a 40.000

componentes ≥ 40.000

**entonces:** # tiene un descuento

**Resultado**: 20%

**Si:**

cliente = COMMAQ

**entonces**: # se descuenta un 2% del 20%

**Resultado**: 18%

**Si no**: # cliente es BEL

**entonces**: # se le añade un 1% al 20%

**Resultado**: 21%

**Postcondición:**

componentes < 10.000 y cliente = COMMAQ => Resultado: 0%

componentes < 10.000 y cliente = BEL => Resultado: 0%

componentes < 20.001 y cliente = COMMAQ => Resultado: 8%

componentes < 20.001 y cliente = BEL => Resultado: 11%

componentes < 40.000 y cliente = COMMAQ => Resultado: 13%

componentes < 40.000 y cliente = BEL => Resultado: 17%

componentes ≥ 40.000 y cliente = COMMAQ => Resultado: 18%

componentes ≥ 40.000 y cliente = BEL => Resultado: 21%

**Ejercicio 7:**

**Entrada:**

Num\_alumnos: int #el numero de alumnos que van a la excursión

Num\_dias: int #numero de días que se van de excursión

**Precondiciones:**

Num\_alumnos ≥ 0

**Realización:**

Si num\_alumnos > 25

Coste\_trayecto(real) = 61,00

Si num\_alumnos ≤25

Coste\_trayecto(real) = 67,30

Coste\_comida(real) = 3,50 \* num\_dias \* num\_alumnos

Si num\_alumnos > 35

Coste\_alojamiento(real) = 3,50 \* num\_dias \* num\_alumnos

Si 31 ≤ num\_alumnos ≤ 35

Coste\_alojamiento(real) = 4 \* num\_dias \* num\_alumnos

Si num\_alumnos < 35

Coste\_alojamiento(real) = 4,75 \* num\_dias \* num\_alumnos

Coste\_alumno(real) = (coste\_trayecto / num\_alumnos) + (coste\_comida / num\_alumnos) + (coste\_alojamiento / num\_alumnos)

Coste\_total(real) = coste\_trayecto + coste\_comida + coste\_alojamiento

**Resultado:**

Coste\_alumno: real #coste individual de cada alumno

Coste\_total: real #coste total de la excursion

**Ejercicio 8**

**Algoritmo**: Prima anual por conductor

**Entrada:**

distancia (real) # la prima que se recibe por kilómetro al año

antigüedad (tiempo) # la prima que se recibe a los cuatro años

responsable (decimal) # responsable del accidente

accidentes (entero) # número de accidentes

km (decimal) # número de kilómetros que recorre al año

**Resultado**: decimal

**Precondición:**

400 ≥ distancia ≥ 0

antigüedad ≥ 0

responsable ≥ 0

accidentes ≥ 0

km ≥ 0

**Realización**:

**Si**:

responsable > 20% # es responsable del accidente

**Si**: # Número de accidentes 1

accidentes < 2 # número de accidentes responsable

**entonces:**

**Resultado**: prima / 2

**Si no**:

**Si:** # Número de accidentes 2

accidentes < 3 # número de accidentes responsable

**entonces:**

**Resultado**: prima / 3

**Si no**:

**Si:** # Número de accidentes 3

accidentes < 4 # número de accidentes responsable

**entonces:**

**Resultado:** prima / 4

**Si no:** # Número de accidentes mayor que 3

entonces:

**Resultado**: 0

**Si no**:

responsable ≤ 20%

**Resultado:** prima

**Si:**

antigüedad < 4:

**entonces:**

**Resultado**: 0

**Si no:**

**Si:**

antigüedad > 4

**entonces:**

**Resultado:** 20,00 × (antigüedad - 4)

**Si no:**

**entonces:**

**Resultado:** 200

**Postcondición:**

distancia = 0,0167€ × km

antigüedad < 4:

responsable > 20%:

accidentes < 2 => **Resultado**: prima / 2 + distancia

accidentes < 3 => **Resultado**: prima / 3 + distancia

accidentes < 4 => **Resultado**: prima / 4 + distancia

accidentes ≥ 4 => **Resultado**: 0

responsable ≤ 20% => **Resultado**: prima + distancia

antigüedad > 4:

responsable > 20%:

accidentes < 2 => **Resultado**: prima / 2 + distancia + 20,00 × (antigüedad - 4)

accidentes < 3 => **Resultado**: prima / 3 + distancia + 20,00 × (antigüedad - 4)

accidentes < 4 => **Resultado**: prima / 4 + distancia + 20,00 × (antigüedad - 4)

accidentes ≥ 4 => **Resultado**: 0

responsable ≤ 20% => **Resultado**: prima + distancia + 20,00 × (antigüedad - 4)

antigüedad = 4:

responsable > 20%:

accidentes < 2 => **Resultado**: prima / 2 + distancia + 200

accidentes < 3 => **Resultado**: prima / 3 + distancia + 200

accidentes < 4 => **Resultado**: prima / 4 + distancia + 200

accidentes ≥ 4 => **Resultado**: 0

responsable ≤ 20% => **Resultado**: prima + distancia + 200