



Fundamentos de programación, introducción a la algoritmia.

01.11.2020

Matilde Elena Pascual Soto

DAM

conv.1020 (octubre 2020)

Indice

1. Indicar si son correctas o no las expresiones indicadas, especificando el motivo o motivos, dadas las definiciones de tipos, constantes y variables siguientes:

tipo

continente = { AMERICA, EUROPA, ASIA, AFRICA, OCEANIA };

país = { ECUADOR, FRANCIA, CHINA, AUSTRALIA, MARRUECOS };

ftipo

const

numContinentes : entero = 5;

numPaises : entero = 5;

maxDistancia : real = 50.0;

fconst

var

fin: boolea ;

d, q: entero ;

m, w: real ;

v: caracter ;

c, f: continente;

p, a, s: país;

fvar

Expresiones:

a) falso y fin o $p = a$

b) $0 < d$ y $d < \text{maxDistancia}$

c) $\text{codigoACaracter}(\text{numContinentes} + \text{realAEntero}(\text{maxDistancia}))$

d) $a = \text{AUSTRALIA} \vee f = \text{FRANCE} \vee m = \text{maxDistancia} + \text{enterAReal}(\text{caracterACodigo}(v) + \text{numContinentes} + \text{numPaises} \div 3)$

2. Calcular el resultado de las expresiones indicadas, dadas las definiciones de tipos, constantes y variables del ejercicio anterior y suponiendo que las variables tienen los siguientes valores:

$d := 5; q := 2; m := 0.5; w := 20.0; v := 'A'; \text{fin} := \text{falso}; c := f; p := \text{CHINA};$

Expresiones:

a) $\text{enteroAReal}(d) < m < \text{fin} \vee \text{FRANCIA} = p$

b) $'B' = \text{codigoACaracter}(\text{realAEntero}(m) + \text{caracterACodigo}(v)) \vee c \neq f$

c) $-(d * q) \div \text{caracterACodigo}(v) + \text{realAEntero}(w * m) \div$

$\text{caracterACodigo}('A')$

3. En el siguiente algoritmo falta definir las secciones de constantes y variables. Definir correctamente el siguiente algoritmo, en el que falta definir las secciones de constantes y variables, según lo que pueda deducir del mismo. El objeto D tiene un valor fijo (supondremos que corresponde al número 1,2).

$\text{dolar} := 0.0;$

$\text{min} := 0.0;$

$\text{max} := \text{min};$

$\text{dolar} := \text{enteroAReal}(c);$

mientras $\text{dolar} > 0.0$ hacer

$\text{euro} := \text{dolar} * D;$

$t := t + \text{euro};$

s i $\text{euro} > \text{max}$ entonces

```

entonces

max:= euro;

fsi

si euro < min entonces

min:= euro;

fsi

c:= c - 10;

dolar:= enteroAReal( c );

fmientras

m := (min + max) / 2.0;

b := m = dolar;

{ Una vez realizados los cálculos, se muestran los resultados por pantalla. }

falgoritmo

```

4. Construir un algoritmo que cuente los enteros positivos inferiores a 1000 que cumplan que su dígito de mayor peso sea igual a la suma del resto de dígitos. A partir del esqueleto del algoritmo siguiente, rellenar los cuadros grises de manera que el algoritmo resuelva el problema.

1. Indicar si son correctas o no las expresiones indicadas, especificando el motivo o motivos, dadas las definiciones de tipos, constantes y variables siguientes:

a) falso y fin o p = a

Esta expresión es correcta cumple tanto semánticamente como logarítmicamente.

falso y fin o pais = pais

falso y booleano o pais = pais

Obtendremos la siguiente expresión:

Sustituyendo booleano por **cierto** por ejemplo tenemos:

Falso y cierto que nos dará como resultado **Falso**

Sustituyendo booleano por **falso** tenemos:

Falso y falso no dará como resultado **falso**

pais = pais no dará un valor **cierto**

Nos quedaría la expresión: **falso o cierto= cierto**

donde p = a

b) $0 < d$ y $d < \text{maxDistancia}$

$0 < \text{entero}$ y $d < \text{maxDistancia}$

$0 < \text{entero}$ y $\text{entero} < 50.0$

Esta expresión no es correcta semánticamente debido a que mezcla dos datos distintos en la última comparación. **$0 < \text{entero}$** donde 0 si puede ser menor que un entero, pero entero no puede ser menor que un real **$\text{entero} < 50.0$** .

El resultado de evaluar esta expresión es un booleano:

$0 < \text{entero} = \text{cierto}$

$\text{entero} < 50.00 = \text{falso}$

Cierto y falso= falso

$0 < d$ y $d < \text{maxDistancia}$ no se cumple

c) `codigoACaracter(numContinentes + realAEntero(maxDistancia))`

Esta expresión es correcta.

`codigoACaracter(numContinentes + realAEntero(50.0))`

`codigoACaracter(numContinentes + 50)`

`codigoACaracter(5+ 50)`

`codigoACaracter(55)`

"7"

Con esta expresión obtenemos el carácter en la posición 55 de la tabla ASCII el cual es: **"7"**,

d) `a = AUSTRALIA o f = FRANCE o m = maxDistancia + enterAReal(caracterACodigo(v) + numContinentes + numPaises div 3)`

Esta expresión es correcta

`a = AUSTRALIA o f = FRANCE o m = maxDistancia + enterAReal(caracterACodigo(v) + numContinentes + 5 div 3)`

a = AUSTRALIA o f = FRANCE o m = maxDistancia + enteroAReal(caracterACodigo(v) + 5 + 1)

a = AUSTRALIA o f = FRANCE o m = maxDistancia + enteroAReal(caracterACodigo(v) +6)

a = AUSTRALIA o f = FRANCE o m = maxDistancia + enteroAReal(118 +6)

a = AUSTRALIA o f = FRANCE o m = maxDistancia + enteroAReal(124)

a = AUSTRALIA o f = FRANCE o m = maxDistancia +124.0

a = AUSTRALIA o f = FRANCE o m = 50.0 +124.0

a = AUSTRALIA o f = FRANCE o m =174.0

Reemplazando todas las variables por sus valores originales tenemos:

a = AUSTRALIA o f = continente o m =real

En esta expresión se cumple a = AUSTRALIA y m =174.0

Donde m= real, Pero no se cumple f= FRANCE, debido a que f=continente.

Nos daría un booleano que resuelto se vería así:

a = AUSTRALIA= **cierto**

f = continente= **falso**

Cierto o falso= cierto

m =real

Cierto o falso= cierto

2. Calcular el resultado de las expresiones indicadas, dadas las definiciones de tipos, constantes y variables del ejercicio anterior y suponiendo que las variables tienen los siguientes valores:

d := 5; q := 2; m := 0.5; w := 20.0; v := 'A'; fin := falso ; c := f; p := CHINA;

Expresiones:

a) enteroAReal(d) < m < fin o FRANCIA = p

enteroAReal(d) < m < fin o CHINA = p

enteroAReal(d) < m < falso o CHINA = p

$\text{enteroAReal}(d) < 0.5 < \text{falso} \text{ o } \text{CHINA} = p$

$5.0 < 0.5 < \text{falso} \text{ o } \text{FRANCIA} = p$

Si resolvemos la siguiente expresión nos daría un booleano:

5.0 no es menor que $0.5 = \text{falso}$

falso no es menor que falso , sino $\text{falso} = \text{falso}$

CHINA no es igual a p , sino $= \text{FRANCIA}$, lo que también lo convierte en falso .

$\text{falso} < \text{falso} = \text{falso}$

$\text{falso} \text{ o } \text{falso} = \text{falso}$

Me dará como resultado un booleano.

falso

b) 'B' = codigoACaracter(realAEntero(m) + caracterACodigo(v)) o c ≠ f

'B' = codigoACaracter(realAEntero(m) + caracterACodigo("A")) o c ≠ f

'B' = codigoACaracter(realAEntero(m) + 65) o c ≠ f

'B' = codigoACaracter(realAEntero(0.5) + 65) o c ≠ f

'B' = codigoACaracter(0 + 65) o c ≠ f

'B' = codigoACaracter(65) o c ≠ f

'B' = "A" o c ≠ f

$66 = 65 \text{ o } \text{false}$

Me dará como resultado un booleano.

false o false = false

c) $-(d * q) \text{ div } \text{caracterACodigo}(v) + \text{realAEntero}(w * m) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } \text{caracterACodigo}(v) + \text{realAEntero}(20.0 * 0.5) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } \text{caracterACodigo}(v) + \text{realAEntero}(10.0) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } \text{caracterACodigo}(v) + (10) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } \text{caracterACodigo}("A") + (10) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } (65) + (10) \text{ div } \text{caracterACodigo}('A')$

$-(d * q) \text{ div } (65) + (10) \text{ div } 65$

$-(5 * 2) \text{ div } (65) + (10) \text{ div } 65$

$-10 \text{ div } 65 + 10 \text{ div } 65 = 0$

Me dará como resultado un número entero al ser divisiones con signos opuestos.

3. En el siguiente algoritmo falta definir las secciones de constantes y variables. Definir correctamente el siguiente algoritmo, en el que falta definir las secciones de constantes y variables, según lo que pueda deducir del mismo. El objeto D tiene un valor fijo (supondremos que corresponde al número 1,2).

algoritmo euros

const

D float = 1, 2 ;

fconstant

Var dolar: float;

max: float;

min: float;

euro: float;

t: int;

C: int;

m: float;

b: float;

fvar

dolar: 0.0;

euro: 0.0;

t: 0;

max= 0.0 ;


```
c = 1;  
m= 0.0;  
b= 0.0;
```

mientras dolar > 0.0 hacer

```
euro = dolar * D;  
t = t + euro;  
si euro > max entonces  
    max= euro;  
fsi  
si euro < min entonces  
    min= euro;  
fsi  
c= c - 10;  
dolar= enteroAReal( c );
```

fmientras

```
m = (min + max) / 2.0;  
b = m = dolar;
```

{ Una vez realizados los cálculos, se muestran los resultados por pantalla. }

4. Construir un algoritmo que cuente los enteros positivos inferiores a 1000 que cumplan que su dígito de mayor peso sea igual a la suma del resto de dígitos. A partir del esqueleto del algoritmo siguiente, rellenar los cuadros grises de manera que el algoritmo resuelva el problema.

a) **algoritmo** digitos

var

n, contador : **entero** ;

centenas, decenas, unidades: **entero** ;

dígitos : **booleano** ;

fvar

{Pre: cierto }

contador :=

0

n := 1 ;

mientras

n<1000

hacer

si n < 10 **entonces**

digitos := **falso** ;

sino

si n < 100 **entonces**

digitos := n **div** 10 = n **mod** 10;

sinó

Si n<1000 entonces

centena=n//100

decena=n%100//10

unidad=n%100-decena*10

digitos=(decena+unidad==centena)

fsi

fsi

si digitos **entonces**

contador=contador+1

fsi

n := n + 1;

fmientras

Post: {contador contiene el número de enteros positivos inferiores a 1000, que cumplen que su dígito de mayor peso es igual a la suma del resto de dígitos.}

falgoritmo

Nótese que en el algoritmo anterior tiene que rellenar los cuadros con una única instrucción o expresión, a excepción de la rama alternativa del si ... sino, que puede estar constituida por una composición secuencial de acciones. b) Vuelve a escribir el mismo algoritmo pero esta vez utilizando la estructura.

para . .. fpara

en lugar de

mientras ... fmientras.

b) algoritmo digitos

var

n, contador : **entero** ;

centenas, decenas, unidades: **entero** ;

dígitos : **booleano** ;

fvar

{Pre: cierto }

contador := 0

para n=1 **hasta** 1000[**paso** 1] **hacer**

si n < 10 **entonces**

digitos := **falso** ;

sino

si n < 100 **entonces**

digitos := n **div** 10 = n **mod** 10;

sinó

Si n<1000 entonces

centena=n//100

decena=n%100//10

unidad=n%100-decena*10

digitos=(decena+unidad==centena)

fsi

fsi

si digitos **entonces**

contador=contador+1

fsi

fpara