

Capítulo 7

7.6 Ejercicios resueltos (Continuación...)

Ejercicio 7.6.3

Elaborar un algoritmo que lea 20 números en un arreglo de 20 elementos, que los imprima y que permita calcular e imprimir la media, la varianza, o la desviación estándar, o las tres al mismo tiempo, ofreciendo el siguiente menú:

MEDIDAS ESTADISTICAS
1. MEDIA 2. VARIANZA 3. DESVIACION ESTANDAR 4. TODAS 5. FIN
ESCOGER OPCION:

MEDIA se obtiene mediante la sumatoria de todos los números dividida entre la cantidad de números (20).

VARIANZA se obtiene la sumatoria del cuadrado de cada una de las desviaciones de cada número respecto a la media, ésta sumatoria se divide entre la cantidad de números (20) menos 1 (N-1; si la cantidad de números es N).

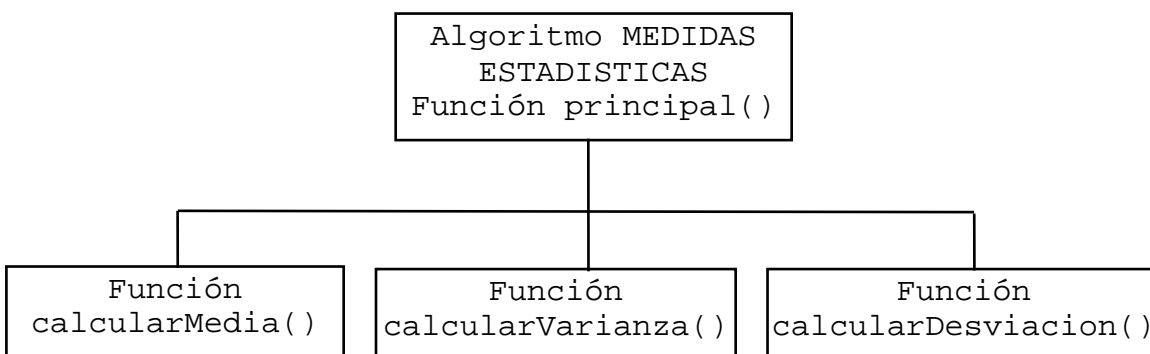
DESVIACION ESTANDAR se obtiene mediante la raíz cuadrada de la varianza.

Usar variables globales.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo MEDIDAS ESTADISTICAS

1. Declarar

Variables

numeros: Arreglo[20] Real

n, opcion: Entero

sumatoria, media, varianza, desviacion,

desviacionEstandar: Real

2. Función principal()

a. Imprimir MENU

MEDIDAS ESTADISTICAS
1. MEDIA 2. VARIANZA 3. DESVIACION ESTANDAR 4. TODAS 5. FIN
ESCOGER OPCION:

b. Leer opcion

c. for n=0; n<=19; n++

1. Solicitar Numeros[n]

2. Leer numeros[n]

d. endfor

e. for n=0; n<=19; n++

1. Imprimir numeros[n]

f. endfor

g. if (opcion==1)OR(opcion==4) THEN

1. calcularMedia()

2. Imprimir media

h. endif

i. if (opcion==2)OR(opcion==4) THEN

1. calcularVarianza()

2. Imprimir varianza

j. endif

k. if (opcion==3)OR(opcion==4) THEN

1. calcularDesviacion()

2. Imprimir desviacionEstandar

l. endif

m. Fin Función principal

3. Función calcularMedia()

a. sumatoria = 0

```

b. for n=0; n<=19; n++
    1. sumatoria = sumatoria + numeros[n]
c. endfor
d. media = sumatoria / n
e. Fin Función calcularMedia

4. Función calcularVarianza()
a. calcularMedia()
b. sumatoria = 0
c. for n=0; n<=19; n++
    1. desviacion = numeros[n] - media
    2. desviacion = Potencia(desviación, 2)
    3. sumatoria = sumatoria + desviacion
d. endfor
e. varianza = sumatoria / (n-1)
f. Fin Función calcularVarianza

5. Función calcularDesviacion()
a. calcularVarianza()
b. desviacionEstandar = RaízCuad(varianza)
c. Fin Función calcularDesviacion
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C712.C
Programa en Java: MedidasEstadisticas.java

Explicación:

En el paso 1 se declaran las variables globales; luego se definen cuatro funciones: Una es la función principal(); y las otras funciones, calculan la media, la varianza y la desviación estándar, respectivamente:

```

1. Función principal()
a. Imprime MENU y solicita la opción
b. Lee opcion
c. Inicia ciclo for desde n = 0 hasta 19
    1. Solicita numeros[n]
    2. Lee numeros[n]
d. Fin del for
e. Inicia ciclo desde n = 0 hasta 19
    1. Imprime numeros[n]
f. Fin del for
g. Si (opcion==1)OR(opcion==4) Entonces
    1. Llama calcularMedia
    2. Imprime media
h. Fin del if
i. Si (opcion==2)OR(opcion==4) Entonces
    1. Llama calcularVarianza

```

- 2. Imprime varianza
 - j. Fin del if
 - k. Si (opción==3)OR(opción==4) Entonces
 - 1. Llama calcularDesviacion
 - 2. Imprime desviacionEstandar
 - l. Fin del if
 - m. Fin Función principal
2. Función calcularMedia()
- a. Inicia sumatoria en 0
 - b. Inicia ciclo for desde n = 0 hasta 19
 - 1. Incrementa Sumatoria con numeros[n]
 - c. Fin del for
 - d. Calcula media = sumatoria / n
 - e. Fin Función calcularMedia
3. Función calcularVarianza()
- a. Llama calcularMedia
 - b. Inicia sumatoria en 0
 - c. Inicia ciclo for desde n = 0 hasta 19
 - 1. calcula desviacion = numeros[n] - media
 - 2. Eleva desviacion al cuadrado
 - 3. Incrementa sumatoria con desviacion
 - d. Fin del for
 - e. Calcula varianza = sumatoria / (n-1)
 - f. Fin Función calcularVarianza
4. Función calcularDesviacion()
- a. Llama calcularVarianza
 - b. Calcula desviacionEstandar = RaízCuad(varianza)
 - c. Fin Función calcularDesviacion
- Fin del algoritmo

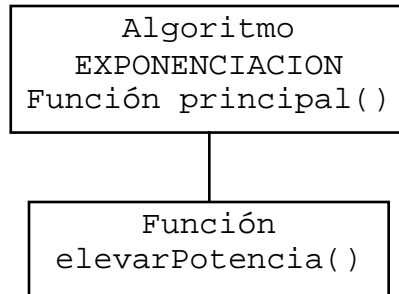
Ejercicio 7.6.4

Elaborar un algoritmo que permita leer la base y el exponente, que llame a una Función que eleve la base al exponente, al regresar a la Función principal que imprima el resultado. Usar parámetros por valor.

A continuación se presenta el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo EXPONENCIACION

1. Función principal()
 - a. Declarar
 - Variables
 - base, exponente, resultado: Real
 - b. Solicitar Base, Exponente
 - c. Leer base, exponente
 - d. resultado = elevarPotencia(base, exponente)
 - e. Imprimir resultado
 - f. Fin Función principal

 2. Función elevarPotencia(Val b, e: Real)
 - a. Declarar
 - Variables
 - poten: Real
 - b. poten = Exp(e * Ln(b))
 - c. return poten
 - d. Fin Función elevarPotencia
- Fin

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C713.C

Programa en Java: Exponenciacion.java

Explicación:

Se define la Función elevarPotencia() que recibe en los parámetros por valor b y e los valores de la base y el exponente, calcula y devuelve el resultado de elevar la base a la potencia utilizando las funciones Exp y Ln. También podría hacerse así: Potencia(b,e).

En la función principal() se lee la base y el exponente; luego se utiliza la función elevarPotencia(), enviándose dichos datos como parámetros por valor, por último se imprime el resultado.

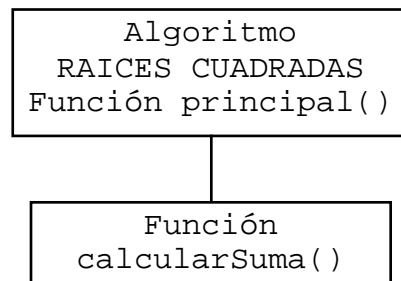
Ejercicio 7.6.5

Elaborar un algoritmo que imprima la sumatoria de las raíces cuadradas de los números pares desde 2 hasta 1500. Utilizar un Función para el cálculo de la sumatoria. Usar parámetro por referencia.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo RAICES CUADRADAS

```
1. Función principal()
  a. Declarar
     Variables
     sumatoria: Real
  b. calcularSuma(sumatoria)
  c. Imprimir sumatoria
  d. Fin Función principal

2. Función calcularSuma(Ref suma: Real)
  a. Declarar
     Variables
     r: Entero
  b. suma = 0
     r = 0
  c. do
     1. r = r + 2
     2. suma = suma + RaízCuad(r)
  d. while r != 1500
  e. Fin Función calcularSuma

Fin
```

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C714.C

Programa en Java: RaicesCuadradas.java

Explicación:

1. Función principal()
 - a. Se declara la variable sumatoria
 - b. Llama calcularSuma(sumatoria) enviando el parámetro sumatoria
 - c. Imprime sumatoria
 - d. Fin Función principal

2. Función calcularSuma(Ref suma: Real)

suma se define como parámetro por referencia.

 - a. Declara la variable r
 - b. Inicia suma en 0
Inicia r en 0
 - c. Inicia ciclo do
 1. Incrementa r con 2
 2. Incrementa suma con RaízCuad(r)
 - d. Fin del ciclo while r !=1500 regresa al do
 - e. Fin Función calcularSuma

Fin del algoritmo

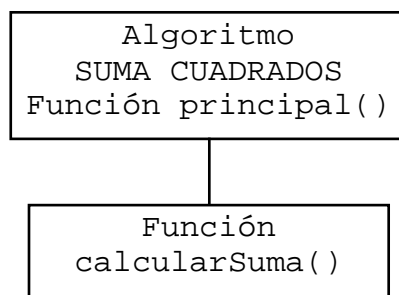
Ejercicio 7.6.6

Elaborar un algoritmo que lea un número N, que imprima la sumatoria de los cuadrados de los enteros entre 1 y N. Utilizar una función para el cálculo de la sumatoria vía parámetro por referencia. También usar parámetro por valor.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

```
Algoritmo SUMA CUADRADOS
1. Función principal()
  a. Declarar
```

```

    Variables
    n: Entero
    sumatoria: Real
b. Solicitar N
c. Leer n
d. calcularSuma(sumatoria,n)
e. Imprimir sumatoria
f. Fin Función principal

2. Función calcularSuma(Ref suma: Real; Val x: Entero)
a. Declarar
    Variables
    r: Entero
b. suma = 0
   r = 0
c. do
    1. r = r + 1
    2. suma = suma + Potencia(r,2)
d. while r != x
e. Fin Función calcularSuma
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C715.C
Programa en Java: SumaCuadrados.java

Explicación:

1. Función principal()
 - a. Se declaran las variables
 - b. Solicita N
 - c. Lee n
 - d. Llama calcularSuma(sumatoria,n) enviando los parámetros sumatoria y n.
 - e. Imprime sumatoria
 - f. Fin Función principal

2. Función calcularSuma(Ref suma: Real; Val x: Entero)

Declara suma como parámetro por referencia; y, x por valor.

 - a. Se declaran las variables
 - b. Inicia suma en 0

Inicia r en 0
 - c. Inicia ciclo do
 1. Incrementa r con 1
 2. Incrementa suma con el cuadrado de r
 - d. Fin ciclo while r != x regresa al do
 - e. Fin Función calcularSuma

Fin del algoritmo

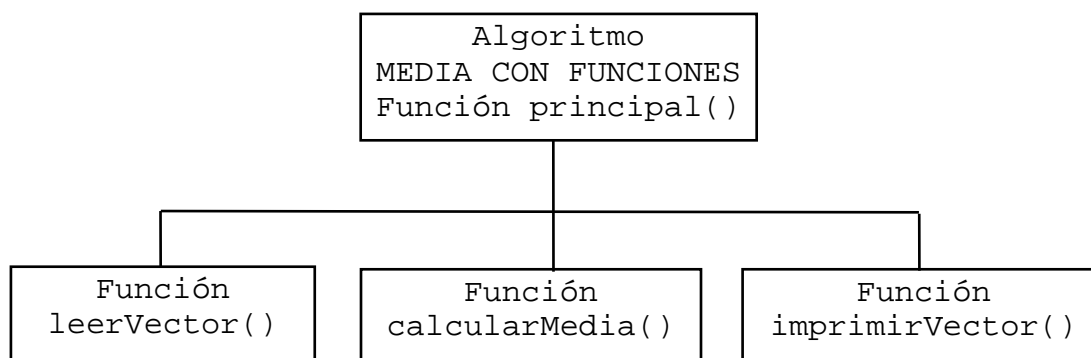
Ejercicio 7.6.7

Elaborar un algoritmo que permita leer diez números de tipo real en un arreglo de 10 elementos en una función, utilizar otra función que reciba como parámetro el arreglo y calcule la media (promedio) e imprima el resultado (en otra función). Usar un arreglo de una dimensión como parámetro.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo MEDIA CON FUNCIONES

1. Función principal()
 - a. Declarar Variables
 - vector: Arreglo[10] Real
 - promedio: Real
 - b. leerVector(vector)
 - c. promedio = calcularMedia(vector)
 - d. imprimirVector(vector)
 - e. Imprimir promedio
 - f. Fin Función principal

2. Función leerVector(Ref vec: Arreglo[10] Real)
 - a. Declarar Variables
 - n: Entero
 - b. for n=0; n<=9; n++
 1. Solicitar elemento vec[n]
 2. Leer vec[n]
 - c. endfor
 - d. Fin Función leerVector

3. Función calcularMedia(Ref v: Arreglo[10] Real) Real
 - a. Declarar
 - Variables
 - sumatoria, prom: Real
 - i: Entero
 - b. sumatoria = 0
 - c. for i=0; i<=9; i++
 1. sumatoria = sumatoria + v[i]
 - d. endfor
 - e. prom = sumatoria / i
 - f. return prom
 - g. Fin Función calcularMedia

4. Función imprimirVector(Ref vect: Arreglo[10] Real)
 - a. Declarar
 - Variables
 - x: Entero
 - b. for x=0; x<=9; x++
 1. Imprimir vect[x]
 - c. endfor
 - d. Fin Función imprimirVector

Fin

En la zona de descarga de la Web del libro, están disponibles:
 Programa en C: C716.C
 Programa en Java: MediaConMetodos.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - vector: Arreglo de 10 elementos
 - promedio: Real
 - b. Llama leerVector(vector) enviando vector como parámetro
 - c. Llama la función calcularMedia(vector) enviando vector como parámetro; y coloca en promedio el valor que devuelve.
 - d. Llama imprimirVector(vector) enviando vector como parámetro
 - e. Imprime promedio
 - f. Fin Función principal

2. Función leerVector(Ref vec: Arreglo de 10 elementos)

Declara vec como parámetro por referencia, para conectarlo con vector.

 - a. Se declara la variable n
 - b. Inicia ciclo for desde n = 0 hasta 9
 1. Solicita elemento vec[n]
 2. Lee vec[n]
 - c. Fin del for

d. Fin Función leerVector

3. Función calcularMedia(Ref v: Arreglo de 10 elementos) Real

Declara v como parámetro por referencia (podría ser por valor), para conectarlo con vector.

- a. Se declaran variables
- b. sumatoria = 0
- c. Inicia ciclo for desde i = 0 hasta 9
 1. Incrementa sumatoria con v[i]
- d. Fin del for
- e. Calcula prom = sumatoria / i
- f. Regresa el valor de prom
- g. Fin Función calcularMedia

4. Función imprimirVector(Ref vect: Arreglo de 10 elementos)

Declara v como parámetro por referencia (podría ser por valor), para conectarlo con vector.

- a. Se declara la variable x
- b. Inicia ciclo for desde x = 0 hasta 9
 1. Imprime vect[x]
- c. Fin del for
- d. Fin Función imprimirVector

Fin del algoritmo

Ejercicio 7.6.8

Elaborar un algoritmo que lea diez valores para Y y X que calcule el valor de Z e imprima los valores de X, Y y Z en cada ocasión; utilizando la función:

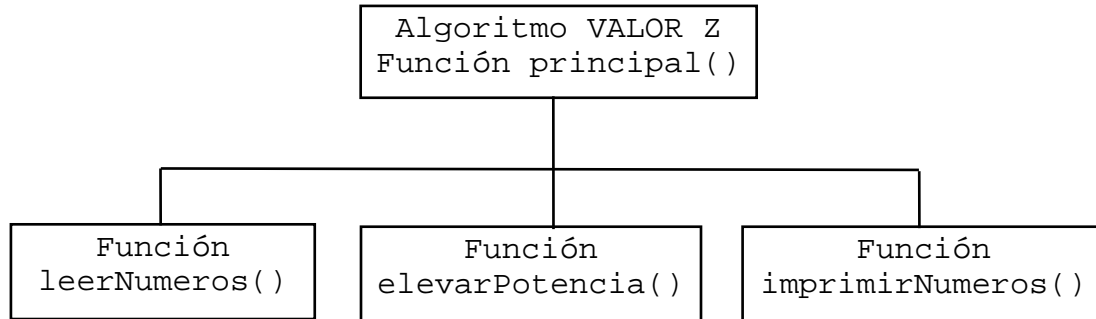
$$Z = Y^X$$

Además del función principal, utilizar una función para leer los valores de X y Y, otra función para elevar Y a la X, y otra función para imprimir los valores de X, Y y Z.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo VALOR Z

1. Función principal()
 - a. Declarar
 - Variables
 - n: Entero
 - x, y, z: Real
 - b. for n=1; n<=10; n++
 1. leerNumeros(x,y)
 2. z = elevarPotencia(y,x)
 3. imprimirNumeros(x,y,z)
 - c. endfor
 - d. Fin Función principal
 2. Función leerNumeros(Ref a, b: Real)
 - a. Solicitar Numero 1, Numero 2
 - b. Leer a, b
 - c. Fin Función leerNumeros
 3. Función elevarPotencia(Val b, e: Real) Real
 - a. Declarar
 - Variables
 - potencia: Real
 - b. potencia = Potencia(b, e)
 - c. return potencia
 - d. Fin Función elevarPotencia
 4. Función imprimirNumeros(Val x2, y2, z2: Real)
 - a. Imprimir x2, y2, z2
 - b. Fin Función imprimirNumeros
- Fin

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C717.C

Programa en Java: ValorZ.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Inicia ciclo for desde $n = 1$ hasta 10
 1. Llama a la Función leerNumeros(x,y); enviándole x y y como parámetros.
 2. $z = \text{elearPotencia}(y,x)$; enviándole x y y como parámetros.
Coloca en z el valor que devuelve el llamado de la Función $\text{elearPotencia}(y,x)$, a la que se le envían los parámetros y y x.
 3. Llama la Función imprimirNumeros(x,y,z); enviándole x, y y z como parámetros.
 - c. Fin del for
 - d. Fin Función principal
 2. Función leerNumeros(Ref a, b: Real)
 - a. Solicita Numero 1, Numero 2
 - b. Lee a, b; los cuales se reflejan en x y y por ser parámetros por referencia.
 - c. Fin Función leerNumeros
 3. Función elevarPotencia(Val b, e: Real) Real
Declara b y e como parámetros por valor, los cuales reciben los valores de y y x.
 - a. Se declara la variable potencia
 - b. Eleva b a la potencia e y coloca el resultado en potencia
 - c. Regresa el valor de potencia
 - d. Fin Función elevarPotencia
 4. Función imprimirNumeros(Val x2, y2, z2: Real)
Declara x2, y2, z2 como parámetros por valor, los cuales reciben los valores de x, y, z.
 - a. Se imprimen los valores de x2, y2, z2
 - b. Fin Función imprimirNumeros
- Fin del algoritmo

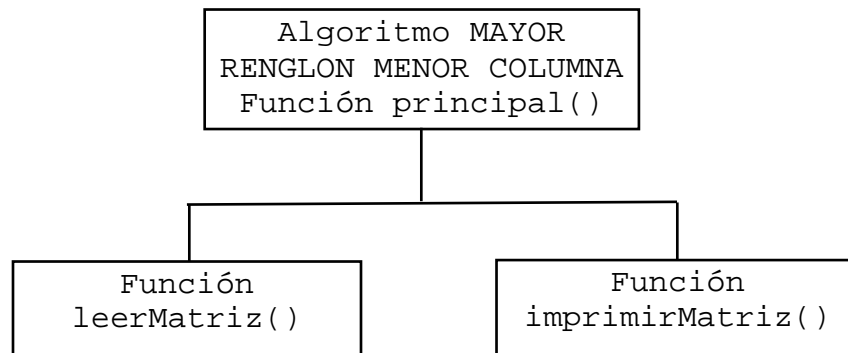
Ejercicio 7.6.9

Elaborar un algoritmo que permita leer números enteros en una matriz de 4X3; imprimir la matriz y el elemento que al mismo tiempo sea el mayor de su renglón y el menor de su columna. Usar un arreglo de dos dimensiones como parámetro.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo MAYOR RENGLON MENOR COLUMNA

1. Función principal()
 - a. Declarar
 - Variables
 - matriz: Arreglo[4][3] Entero
 - b. leerMatriz(matriz)
 - c. imprimirMatriz(matriz)
 - d. Fin Función principal

2. Función leerMatriz(Ref mat: Arreglo[4][3] Entero)
 - a. Declarar
 - Variables
 - r, c: Entero
 - b. for r=0; r<=3; r++
 1. for c=0; c<=2; c++
 - a. Solicitar elemento mat[r][c]
 - b. Leer mat[r][c]
 2. endfor
 - c. endfor
 - d. Fin Función leerMatriz

3. Función imprimirMatriz(Ref ma: Arreglo[4][3] Entero)
 - a. Declarar
 - Variables
 - re, co, r, c, ren, mayRen, menCol, reMa, coMa, reMe, coMe: Entero
 - b. for re=0; re<=3; re++
 1. for co=0; co<=2; co++
 - a. Imprimir ma[re][co]
 2. endfor
 - c. endfor
 - d. for ren=0; ren<=3; ren++

```

1. mayRen = ma[ren,0]
   reMa = ren
   coMa = 0
2. for c=1; c<=2; c++
   a. if ma[ren][c] > mayRen THEN
       1. mayRen = ma[ren][c]
       2. reMa = ren
       3. coMa = c
   b. endif
3. endfor
4. menCol = ma[0,coMa]
   reMe = 1
   coMe = coMa
5. for r=1; r<=3; r++
   a. if ma[r][coMa] < menCol THEN
       1. menCol = ma[r][coMa]
       2. reMe = r
       3. coMe = coMa
   b. endif
6. endfor
7. if (reMa == reMe)AND(coMa == coMe) THEN
   a. Imprimir ma[reMa][coMa], reMa, coMa
8. endif
e. endfor
f. Fin Función imprimirMatriz
Fin

```

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C718.C

Programa en Java: MayorRenMenorCol.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Llama la Función leerMatriz(matriz); envía matriz como parámetro.
 - c. Llama la Función imprimirMatriz(matriz); envía matriz como parámetro.
 - d. Fin Función principal

2. Función leerMatriz(Ref mat: Arreglo[4][3] Entero)

Declara mat como parámetro por referencia. Para conectarla con matriz

 - a. Declara las variables r y c
 - b. Inicia ciclo for desde r = 0 hasta 3
 1. Inicia ciclo for desde c = 0 hasta 2
 - a. Solicita elemento mat[r,c]
 - b. Lee mat[r,c]
 2. Fin del for
 - c. Fin del for

- d. Fin Función leerMatriz
3. Función imprimirMatriz(Ref ma: Arreglo[4][3] Entero)
- a. Declara las variables
 - re, co, r, c, ren, mayRen, menCol, reMa, coMa, reMe, coMe: Entero
 - b. Inicia ciclo for desde re = 0 hasta 3
 - 1. Inicia ciclo for desde co = 0 hasta 2
 - a. Imprime elemento ma[re][co]
 - 2. Fin del for
 - c. Fin del for
 - d. Inicia ciclo for desde ren = 0 hasta 3
 - 1. Coloca en mayRen el elemento ma[ren,0]
Coloca en reMa el valor de ren
Coloca en coMa el valor 0
 - 2. Inicia ciclo for desde c = 0 hasta 2
 - a. Si ma[ren][c] > mayRen Entonces
 - 1. Coloca en mayRen el elemento ma[ren][c]
 - 2. Coloca en reMa el valor de ren
 - 3. Coloca en coMa el valor de c
 - b. Fin del if
 - 3. Fin del for
 - 4. Coloca en menCol el elemento ma[0,coMa]
Coloca en reMe el valor 1
Coloca en coMe el valor de coMa
 - 5. Inicia ciclo for desde r = 1 hasta 3
 - a. Si ma[r][coMa] < menCol Entonces
 - 1. Coloca en menCol el elemento ma[r][coMa]
 - 2. Coloca en reMe el valor de r
 - 3. Coloca en coMe el valor de coMa
 - b. Fin del if
 - 6. Fin del for
 - 7. Si (reMa == reMe)AND(coMa == coMe) Entonces
 - a. Imprime ma[reMa][coMa], reMa, coMa
 - 8. Fin del if
 - e. Fin del for
 - f. Fin Función imprimirMatriz
- Fin del algoritmo

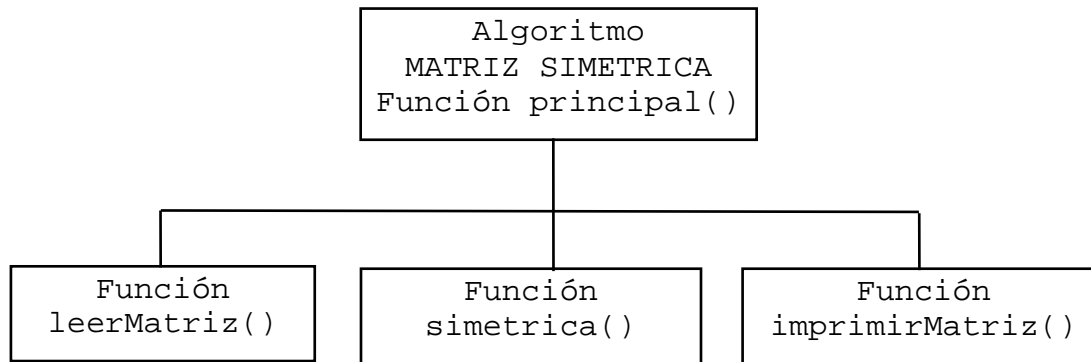
Ejercicio 7.6.10

Elaborar un algoritmo que lea números en una matriz de 4X4 e indique si es simétrica o no; es simétrica si el elemento i,j es igual al elemento j,i para todo par de valores de i y j . Utilizar una función para la lectura mediante parámetro por referencia, y otra función que regrese el valor 1 si es simétrica o 0 si no lo es. Usar un arreglo de dos dimensiones como parámetro.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo MATRIZ SIMETRICA

1. Función principal()

a. Declarar

Variables

matriz: Arreglo[4][4] Entero

resultado: Entero

b. leerMatriz(matriz)

c. resultado = simetrica(matriz)

d. imprimirMatriz(matriz)

e. if resultado == 1 THEN

1. Imprimir "LA MATRIZ ES SIMETRICA"

f. else

1. Imprimir "LA MATRIZ NO ES SIMETRICA"

g. endif

h. Fin Función principal

2. Función leerMatriz(Ref mat: Arreglo[4][4] Entero)

a. Declarar

Variables

r,c: Entero

b. for r=0; r<=3; r++

1. for c=0; c<=3; c++

a. Solicitar elemento mat[r][c]

b. Leer mat[r][c]

2. endfor

c. endfor

d. Fin Función leerMatriz

```

3. Función simetrica(Val m: Arreglo[4][4] Entero) Entero
  a. Declarar
      Variables
          ren, col, resul: Entero
  b. resul = 1
  c. for ren=0; ren<=3; ren++
      1. for col=0; col<=3; col++
          a. if m[ren][col] != m[col][ren] THEN
              1. resul = 0
          b. endif
      2. endfor
  d. endfor
  e. return resul
  f. Fin Función simetrica

4. Función imprimirMatriz(Ref ma: Arreglo[4][4] Entero)
  a. Declarar
      Variables
          re, co: Entero
  b. for re=0; re<=3; re++
      1. for co=0; co<=3; co++
          a. Imprimir ma[re][co]
      2. endfor
  c. endfor
  d. Fin Función imprimirMatriz
Fin

```

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C719.C

Programa en Java: MatrizSimetrica.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Llama leerMatriz(matriz); envía matriz como parámetro.
 - c. Coloca en resultado el valor que devuelva al llamar simetrica(matriz); envía matriz como parámetro.
 - d. Llama imprimirMatriz(matriz) envía matriz como parámetro.
 - e. Si resultado == 1Entonces
 1. Imprime “LA MATRIZ ES SIMETRICA”
 - f. Si no
 1. Imprime “LA MATRIZ NO ES SIMETRICA”
 - g. Fin del if
 - h. Fin Función principal
2. Función leerMatriz(Ref mat: Arreglo[4][4] Entero)

Declara mat como parámetro por referencia. Para conectarla con matriz

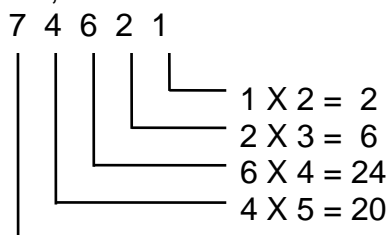
- a. Declara las variables r y c
 - b. c. Lee los elementos de la matriz.
 - d. Fin Función leerMatriz
3. Función simetrica(Val m: Arreglo[4][4] Entero) Entero
 Declara m como parámetro por referencia. Para conectarla con matriz
- a. Declara la variable resul
 - b. Inicia asimetria con 1
 - c. ciclo for desde r = 0 hasta 3
 - 1. ciclo for desde c = 0 hasta 3
 - a. Compara si $m[r,c] \neq m[c,r]$ Si es asi, Entonces
 - 1. Coloca en resul el valor 0
 - b. Fin if
 - 2. Fin for
 - d. Fin for
 - e. Regresa el valor de resul
 - f. Fin Función simetrica
4. Función imprimirMatriz(Ref ma: Arreglo[4][4] Entero)
 Declara ma como parámetro por referencia. Para conectarla con matriz
- a. ciclo for desde r = 0 hasta 3
 - 1. ciclo for desde c = 0 hasta 3
 - a. Imprime $ma[r,c]$
 - 2. Fin for
 - b. Fin for
 - c. Fin Función imprimirMatriz
- Fin del algoritmo

Ejercicio 7.6.11

Elaborar un algoritmo que permita leer un número de tipo entero de cinco cifras; calcular e imprimir el dígito verificador. El dígito verificador es un número entre 0 y 9, que se obtiene a partir de un número en el que se toma en cuenta la posición que tiene cada dígito; se obtiene de la siguiente manera:

Si se tiene el número: 74621

Proceso 1 Se obtiene la suma de los productos: el primer número de la derecha multiplicado por dos, el siguiente a la izquierda por tres, el siguiente por cuatro, y así sucesivamente, hasta siete (el multiplicador vuelve a iniciar en 2, en caso de ser necesario).



$$\begin{array}{r} \text{—————} \\ 7 \times 6 = \underline{42} \\ 94 \end{array}$$

Proceso 2 Se determina el residuo al dividir el producto entre once.

$$\begin{array}{r} 8 \\ 11 \overline{) 94} \\ \underline{6} \end{array}$$

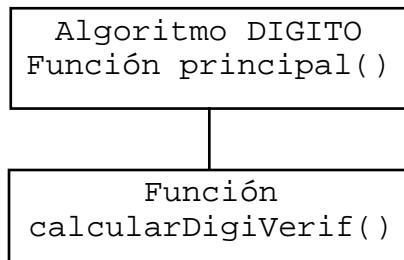
Proceso 3 Si el residuo es cero o uno, entonces éste es el dígito verificador; de lo contrario el digito verificador será igual al resultado que se obtenga de once menos el residuo; en el ejemplo:

$$\text{Digito verificador} = 11 - 6 = 5$$

A continuación se presenta el algoritmo que realiza el proceso explicado anteriormente.

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo DIGITO

1. Función principal()

a. Declarar

Variables

n, digiVerif: Entero

b. Solicitar N

c. Leer n

d. digiVerif = calcularDigiVerif(n)

e. Imprimir digiVerif

f. Fin Función principal

2. Función calcularDigiVerif(Val nume: Entero)

a. Declarar

Variables

i, producto, multiplicador, n2, e, digVeri,

```

        residuo: Entero
        numero: Cadena
        num: Carácter
    b. Cadena( nume, numero)
    c. producto = 0
    d. multiplicador = 1
    e. for i=Longitud(numero)-1; i>=0; i--
        1. multiplicador = multiplicador + 1
        2. num = SubCadena(numero,i,1)
        3. Valor(num,n2,e)
        4. producto = producto + (n2 * multiplicador)
    f. endfor
    g. residuo = producto
    h. while residuo > = 11
        1. residuo = residuo - 11
    i. endwhile
    j. if (residuo == 0) OR (residuo == 1) THEN
        1. digVeri = residuo
    k. else
        1. digVeri = 11 - residuo
    l. endif
    m. return DigVeri
    n. Fin Función calcularDigiVerif
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C720.C
Programa en Java: Digito.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Solicita N
 - c. Lee n
 - d. Coloca en digiVerif = el valor devuelto por calcularDigiVerif(n), envía el parámetro n.
 - e. Imprime digiVerif
 - f. Fin Función principal

2. Función calcularDigiVerif(Val nume : Entero)

Declara nume como parámetro por valor, para conectarse con n.

 - a. Declara variables
 - b. Convierte nume a Cadena y lo coloca en numero
 - c. Inicia producto en 0
 - d. Inicia multiplicador en 1
 - e. Ciclo for desde i = la longitud del numero menos 1, hasta 0 con decrementos de -1
 1. Incrementa multiplicador en 1
 2. Coloca en num la SubCadena(numero,i,1)

3. Convierte a numérico num y lo pone en n2
 4. Incrementa producto con (n2 * multiplicador)
 - f. Fin for
 - g. Coloca en residuo el valor de producto
 - h. Ciclo while residuo >= 11
 1. Le quita 11 a residuo
 - i. Fin while
 - j. Si (residuo == 0) OR (residuo == 1) Entonces
 1. Coloca en digVeri el valor de residuo
 - k. Si no
 1. Coloca en digVeri el resultado de 11 - residuo
 - l. Fin if
 - m. Regresa valor de DigVeri
 - n. Fin Función calcularDigiVerif
- Fin del algoritmo

Ejercicio 7.6.12

Elaborar un algoritmo que permita leer una frase u oración y que imprima si es un palíndromo o no. Un palíndromo es una frase que se lee igual de izquierda a derecha que de derecha a izquierda .

Por ejemplo: RADAR se lee igual de izquierda a derecha que de derecha a izquierda. Usar funciones cadena de caracteres.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

Algoritmo PALINDROMO

1. Función principal()
 - a. Declarar
 - Variables
 - frase, frase2: Cadena
 - car: Carácter
 - l: Entero
 - b. Solicitar la frase
 - c. Leer frase
 - d. frase2 = ""
 - e. for l=Longitud(frase)-1; l>=0; l--
 1. car = SubCadena(frase,l,1)
 2. frase2 = frase2 + car
 - f. endfor
 - g. if frase == frase2 THEN
 1. Imprimir "ES UN PALINDROMO"
 - h. else
 1. Imprimir "NO ES UN PALINDROMO"
 - i. endif

```
j. Fin Función principal
Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C721.C
Programa en Java: Palindromo.java

Explicación:

```
1. Función principal()
  a. Se declaran variables
  b. Solicita frase
  c. Lee frase
  d. Inicia frase2 en nulo
  e. Inicia ciclo for desde l = Longitud(frase) menos 1, hasta 0 con decrementos de -1
    1. Copia el siguiente carácter en car
    2. Concatena en Frase2 el car
  f. Fin for
  g. Si frase == frase2 Entonces
    1. Imprime " ES UN PALINDROMO "
  h. Si no
    1. Imprime " NO ES UN PALINDROMO "
  i. Fin if
  j. Fin Función principal
Fin del algoritmo
```

Ejercicio 7.6.13

Elaborar un algoritmo que lea una fecha en forma DDMMAAAA, es decir,
DD son dos dígitos que indican el día
MM son dos dígitos que indican el mes
AAAA son cuatro dígitos que indican el año.
Imprimir la fecha: Por ejemplo; si tecléa 21082005 debe imprimir:

21 de Agosto de 2005

Usar funciones cadena de caracteres.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

Algoritmo DICE FECHA

```
1. Función principal()
  a. Declarar
    fecha: Cadena[8]
    dia, mes2, anio: Cadena[2]
    anio: Cadena[4]
```

```

        nombreMes: Cadena[15]
        nuMes, x: Entero
b. Solicitar fecha
c. Leer fecha
d. dia = SubCadena( fecha, 1, 2)
   mes2 = SubCadena( fecha, 3, 2)
   Valor( mes2, nuMes, x)
   anio = SubCadena( fecha, 5, 4)
e. switch nuMes
    1: nombreMes = "Enero"
    2: nombreMes = "Febrero"
    3: nombreMes = "Marzo"
    4: nombreMes = "Abril"
    5: nombreMes = "Mayo"
    6: nombreMes = "Junio"
    7: nombreMes = "Julio"
    8: nombreMes = "Agosto"
    9: nombreMes = "Septiembre"
   10: nombreMes = "Octubre"
   11: nombreMes = "Noviembre"
   12: nombreMes = "Diciembre"
f. endswitch
g. Imprimir dia, " de ", nombreMes, " de ", anio
h. Fin Función principal
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C722.C
Programa en Java: DiceFecha.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Solicita fecha
 - c. Lee fecha
 - d. Copia en dia los dos caracteres del día
Copia en mes2 los dos caracteres del mes
Convierte a numérico el mes
Copia en anio los cuatro caracteres del año.
 - e. Inicia switch nuMes
 - Si nuMes == 1: nombreMes = "Enero"
 - Si nuMes == 2: nombreMes = "Febrero"
 - Si nuMes == 3: nombreMes = "Marzo"
 - Si nuMes == 4: nombreMes = "Abril"
 - Si nuMes == 5: nombreMes = "Mayo"
 - Si nuMes == 6: nombreMes = "Junio"
 - Si nuMes == 7: nombreMes = "Julio"

Si nuMes == 8: nombreMes = "Agosto"
 Si nuMes == 9: nombreMes = "Septiembre"
 Si nuMes == 10: nombreMes = "Octubre"
 Si nuMes == 11: nombreMes = "Noviembre"
 Si nuMes == 12: nombreMes = "Diciembre"
 f. Fin switch
 g. Imprime día, " de ", nombreMes, " de ", año
 h. Fin Función principal
 Fin del algoritmo

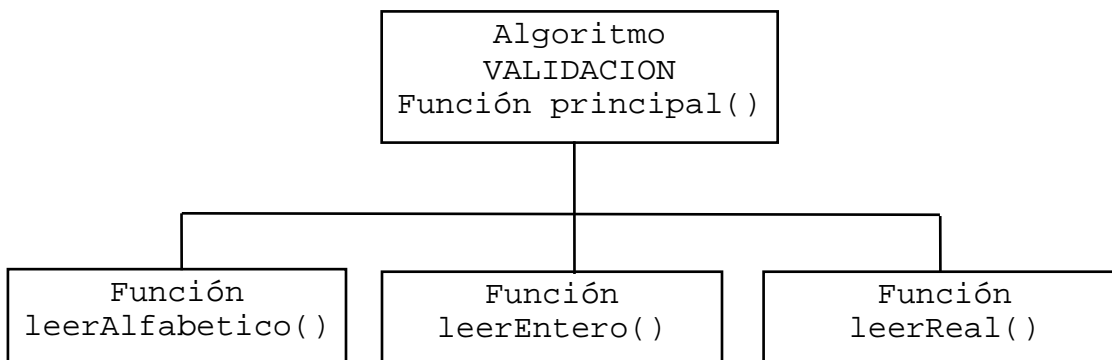
Ejercicio 7.6.14

Elaborar un algoritmo que lea el nombre de un empleado, el número de horas trabajadas y la cuota por hora; hacer y utilizar las funciones leerEntero, la cual permita leer un número de tipo entero normal haciendo la validación, es decir, sólo permitir los dígitos del 0 al 9, el signo de menos y que esté en el rango -32768 al 32767 (tipo entero normal); leerAlfabético, que sólo debe permitir la lectura de caracteres alfabéticos, el espacio y el punto; y leerReal, que sólo debe permitir la lectura de dígitos, el signo de menos y un punto decimal. Imprimir los tres datos leídos. Para leer los datos se deben utilizar funciones que regresan valor, es decir, devuelven el valor leído. Usar funciones de cadenas de caracteres para validar la entrada de datos.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

```

Algoritmo VALIDACION
1. Función principal()
  a. Declarar
    Variables
  
```

```

        nombreEmp: Cadena
        horasTrab: Entero
        cuotaHora: Real
    b. nombreEmp = leerAlfabetico()
    c. horasTrab = leerEntero()
    d. cuotaHora = leerReal()
    e. Imprimir nombreEmp, horasTrab, cuotaHora
    f. Fin Función principal

2. Función leerAlfabetico() Cadena
    a. Declarar
        Variables
        nom: Cadena
        i, bandera: Entero
        car: Carácter
    b. do
        1. bandera = 1
        2. Solicitar Nombre del empleado
        3. Leer nom
        4. for i=0; i<=Longitud(nom)-1; i++
            a. car = SubCadena(nom,i,1)
            b. if NOT(((car >= "a")AND(car<="z"))OR
                ((car>="A")AND(car<="Z"))OR
                (car==" ")OR(car==".")) THEN
                1. bandera = 0
            c. endif
        5. endfor
    c. while bandera != 1
    d. return nom
    e. Fin Función leerAlfabetico

3. Función leerEntero() Entero
    a. Declarar
        Variables
        n, i, e, bandera: Entero
        numero: Cadena
        num: Carácter
    b. do
        1. do
            a. bandera = 1
            b. Solicitar Número de horas trabajadas
            c. Leer numero
            d. for i=0; i<=Longitud(numero)-1; i++
                1. num = SubCadena(numero,i,1)

                2. if NOT(((num>="0")AND(num<="9"))OR
                    (num=="-")) THEN

```

```

        a. bandera = 0
    3. endif
    4. if (num=="-")AND(i>0) THEN
        a. bandera = 0
    5. endif
    e. endfor
2. while bandera != 1
3. n = Valor(numero,n,e)
4. if (n < -32768)OR(n > 32767) THEN
    a. bandera = 0
5. endif
c. while bandera != 1
d. return n
e. Fin Función leerEntero

4. Función leerReal() Real
a. Declarar
    Variables
        z: Real
        numero: Cadena
        i, e, puntos, bandera: Entero
        num: Carácter
b. do
    1. bandera = 1
    2. puntos = 0
    3. Solicitar Cuota por hora
    4. Leer numero
    5. for i=0; i<=Longitud(numero)-1; i++
        a. num = SubCadena(numero,i,1)
        b. if NOT(((num>="0")AND(num<="9"))
            OR(num=="-")OR(num==".")) THEN

                1. bandera = 0
            c. endif
            d. if (num=="-")AND(i>0) THEN
                a. bandera = 0
            e. endif
            f. if num == "." THEN
                1. puntos = puntos + 1
            g. endif
        6. endfor
    7. if puntos > 1 THEN
        a. bandera = 0
    8. endif
c. while bandera != 1
d. n = Valor(numero,z,e)
e. return n

```

f. Fin Función leerReal
Fin

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C723.C
Programa en Java: Validacion.java

Explicación:

1. Función principal()
 - a. Se declaran variables de la Función principal
 - b. Llama leerAlfabetico() y el valor que regresa lo coloca en nombreEmp
 - c. Llama leerEntero() y el valor que regresa lo coloca en horasTrab
 - d. Llama leerReal() y el valor que regresa lo coloca en cuotaHora
 - e. Imprime nombreEmp, horasTrab, cuotaHora
 - f. Fin Función principal

2. Función leerAlfabetico() Cadena
 - a. Se declaran variables
 - b. Inicia ciclo do
 1. Inicia bandera con 1
 2. Solicita Nombre del empleado
 3. Lee nom
 4. Inicia ciclo for desde $i = 0$ hasta $\text{Longitud}(\text{nom})-1$
 - a. Copia en car el siguiente carácter
 - b. Si car no es un carácter válido (letra, espacio o punto); Entonces
 1. Coloca bandera en 0
 - c. Fin if
 5. Fin for
 - c. Fin ciclo while bandera $\neq 1$ (no tecleó caracteres válidos); regresa al do
 - d. Regresa el valor de nom
 - e. Fin Función leerAlfabetico

Repite el ciclo de leer hasta que haya leído un dato alfabético sin errores.

3. Función leerEntero() Entero
 - a. Se declaran variables
 - b. Inicia ciclo do
 1. Inicia ciclo do
 - a. Inicia bandera con 1
 - b. Solicita Número de horas trabajadas
 - c. Lee numero
 - d. Inicia ciclo for desde $i = 0$ hasta 4
 1. Copia en num el siguiente carácter
 2. Si num no es dígito entre 0 y 9; y no es el signo de menos Entonces
 - a. Coloca bandera en 0
 3. Fin if
 4. Si num es menos y no esta en el carácter 0 Entonces
 - a. Coloca bandera en 0

- 5. Fin if
 - e. Fin for
 - 2. Fin ciclo while bandera != 1 (no tecló bien el número); regresa al do
 - 3. Copia en n el siguiente carácter
 - 4. Si (n < -32768)OR(n > 32767) Entonces
 - a. Coloca bandera en 0
 - 5. Fin if
 - c. Fin ciclo while bandera != 1 (no esta en el rango del entero); regresa al do
 - d. Regresa el valor de n
 - e. Fin Función leerEntero
- Repite el ciclo de leer hasta que haya leído un dato entero sin errores.

4. Función leerReal() Real

Declara z como parámetro por referencia.

- a. Se declaran variables
- b. Inicia ciclo do
 - 1. Coloca bandera en 1
 - 2. Inicia puntos en 0
 - 3. Solicitar Cuota por hora
 - 4. Lee numero
 - 5. Inicia ciclo for desde i = 0 hasta Longitud(numero)-1
 - a. Copia en num el siguiente carácter
 - b. Si num no es dígito entre 0 y 9; y no es el signo de menos; ni el punto Entonces
 - 1. Coloca bandera en 0
 - c. Fin if
 - d. Si num es menos y no esta en el carácter 0 Entonces
 - 1. Coloca bandera en 0
 - e. Fin if
 - f. Si num == "." Entonces
 - 1. Incrementa puntos en 1
 - g. Fin if
 - 6. Fin for
 - 7. Si puntos > 1 Entonces
 - a. Coloca bandera en 0
 - 8. Fin if
- c. Fin ciclo while bandera != 1 (no tecló bien el número); regresa al do
- d. Convierte numero a numérico y lo pone en n
- e. Regresa el valor de n
- f. Fin Función leerReal

Repite el ciclo de leer hasta que haya leído un dato real sin errores.

Fin del algoritmo

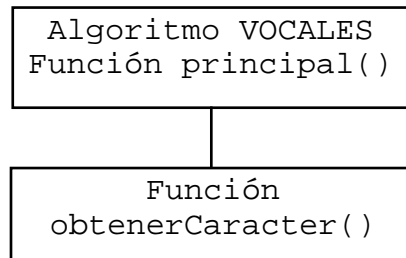
Ejercicio 7.6.15

Elaborar un algoritmo que permita leer una frase u oración e imprimir cuántas veces se utilizó cada una de las vocales. Usar funciones cadena de caracteres y parámetros.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo VOCALES

1. Función principal()
 - a. Declarar
Variables
frase: Cadena
l, a, e, i, o, u: Entero
car: Carácter
 - b. a = 0; e = 0; i = 0; o = 0; u = 0
 - c. Solicitar Frase
 - d. Leer frase
 - e. for l=0; l<=Longitud(frase)-1; l++
 1. car = obtenerCaracter(frase,l)
 2. if (car=='a')OR(car=='A') THEN
 - a. a = a + 1
 3. endif
 4. if (car=='e')OR(car=='E') THEN
 - a. e = e + 1
 5. endif
 6. if (car=='I')OR(car=='I') THEN
 - a. i = i + 1
 7. endif
 8. if (car=='o')OR(car=='O') THEN
 - a. o = o + 1
 9. endif
 10. if (car=='u')OR(car=='U') THEN
 - a. u = u + 1
 11. endif
 - f. endfor
 - g. Imprimir "A=",a, "E=", e, "I=", i, "O=", o, "U=", u
 - h. Fin Función principal

```

2. Función obtenerCaracter(Val fra: Cadena;
                          Ref i: Entero) Caracter
    a. Declarar
        Variables
            carAux: Carácter
    b. carAux = SubCadena(fra,i,1)
    c. return carAux
    d. Fin Función obtenerCaracter

```

Fin

En la zona de descarga de la Web del libro, están disponibles:

Programa en C: C724.C

Programa en Java: Vocales.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Inicia los contadores a = 0; e = 0; i = 0; o = 0; u = 0
 - c. Solicita Frase
 - d. Lee frase
 - e. Inicia ciclo for desde l = 0 hasta Longitud(frase)-1
 1. Obtiene siguiente carácter y lo pone en car
 2. Si (car=='a')OR(car=='A') Entonces
 - a. a = a + 1
 3. Fin if
 4. Si (car=='e')OR(car=='E') Entonces
 - a. e = e + 1
 5. Fin if
 6. Si (car=='i')OR(car=='I') Entonces
 - a. i = i + 1
 7. Fin if
 8. Si(car=='o')OR(car=='O') Entonces
 - a. o = o + 1
 9. Fin if
 10. Si (car=='u')OR(car=='U') Entonces
 - a. u = u + 1
 11. Fin if
 - f. Fin for
 - g. Imprime "A=", a, "E=", e, "I=", i, "O=", o, "U=", u
 - h. Fin Función principal

2. Función obtenerCaracter(Val fra: Cadena100; Ref i: Entero) Caracter

Declara fra como parámetro por valor y por referencia i.

 - a. Declara la variable carAux
 - b. Copia en carAux el siguiente carácter
 - c. Regresa el valor de carAux

d. Fin Función obtenerCaracter
Fin del algoritmo

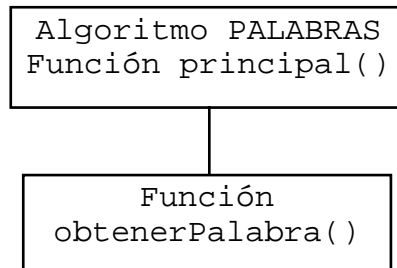
Ejercicio 7.6.16

Elaborar un algoritmo que permita leer una frase u oración, que llame a una función que obtenga una palabra, al regresar a la función principal que la imprima, esto se debe repetir hasta que se termine la frase. Usar funciones cadena de caracteres y parámetros.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo PALABRAS

1. Función principal()
 - a. Declarar
Variables
frase : Cadena
palabra: Cadena
l: Entero
 - b. Solicitar Frase
 - c. Leer frase
 - d. l = 0
 - e. do
 1. palabra = obtenerPalabra(frase,l)
 2. Imprimir palabra
 - f. while l < Longitud(frase)
 - g. Fin Función principal
2. Función obtenerPalabra(Val fra: Cadena;
Ref c: Entero) Cadena
 - a. Declarar
Variables
car: Carácter


```

        pal: Cadena
b. pal = ""
c. do
    2. car = SubCadena(fra,c,1)
    3. if car != ' ' THEN
        a. pal = pal + car
    4. endif
    5. c = c + 1
d. while (car != ' ')AND(c != Longitud(fra))
e. return pal
f. Fin Función obtenerPalabra
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C725.C
Programa en Java: Palabras.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Se solicita la Frase
 - c. Se lee en frase
 - d. Se inicia l en 0
 - e. Inicia do
 1. Obtiene la siguiente palabra en el Función obtenerPalabra(frase,l)
 2. Imprime la palabra
 - f. Fin ciclo while l < Longitud(frase)
 - g. Fin Función principal

 2. Función obtenerPalabra(Val fra: Cadena100; Ref c: Entero) Cadena
 - a. Se declaran variables
 - b. Inicia ciclo do
 1. Inicia pal con nulo
 2. Copia en car el carácter número c de fra
 3. Si car es diferente de espacio; Entonces
 - a. Concatena el car en pal
 4. Fin del if
 5. Incrementa c en 1
 - c. Fin ciclo while (car != ' ')AND(c != Longitud(fra)); Regresa al do
 - d. Regresa el valor de pal
 - e. Fin Función obtenerPalabra
- Fin del algoritmo

Ejercicio 7.6.17

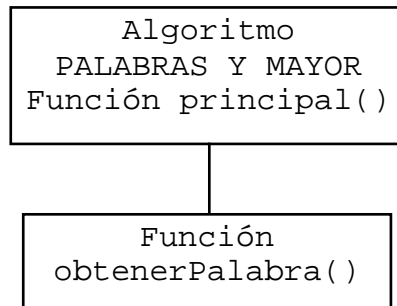
Elaborar un algoritmo que permita leer una frase u oración que llame a una función que obtenga una palabra, al regresar a la función principal que la

imprima, esto se debe repetir hasta que se termine la frase; y al final, que imprima la palabra más larga y su longitud. Supóngase que no habrá dos o más palabras con el mismo tamaño. Usar funciones cadena de caracteres y parámetros.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo PALABRAS Y MAYOR

1. Función principal()
 - a. Declarar
Variables
frase: Cadena
palabra, palabraMayor: Cadena
l, tamañoPal, mayorTamaño: Entero
 - b. Solicitar Frase
 - c. Leer frase
 - d. mayorTamaño = 0
l = 0
 - e. do
 1. palabra = obtenerPalabra(frase,l)
 2. Imprimir palabra
 3. tamañoPal = Longitud(palabra)
 4. if tamañoPal > mayorTamaño THEN
 - a. mayorTamaño = tamañoPal
 - b. palabraMayor = palabra
 5. endif
 - f. while l < Longitud(frase)
 - g. Imprimir palabraMayor, mayorTamaño
 - h. Fin Función principal
2. Función obtenerPalabra(Val fra: Cadena;
Ref c: Entero) Cadena
 - a. Declarar

```

Variables
  car: Carácter
  pal: Cadena
b. pal = ""
c. do
    1. car = SubCadena(fra,c,1)
    2. if car != ' ' THEN
        a. pal = pal + car
    3. endif
    4. c = c + 1
d. while (car != ' ')AND(c != Longitud(fra))
e. return pal
f. Fin Función obtenerPalabra
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
 Programa en C: C726.C
 Programa en Java: PalabrasMayor.java

Explicación:

1. Función principal()
 - a. Se declaran variables
 - b. Se solicita la Frase
 - c. Se lee frase
 - d. Se inicializa mayorTamaño en 0
Se inicializa l en 0
 - e. Inicia ciclo do
 1. Llama obtenerPalabra enviando frase y l como parámetros y la palabra que devuelve la coloca en palabra.
 2. Imprime palabra
 3. Coloca en tamañoPal la longitud de palabra
 4. Si tamañoPal > mayorTamaño Entonces
 - a. Coloca en mayorTamaño el tamañoPal
 - b. Coloca en palabraMayor el valor de palabra
 5. Fin del if
 - f. Fin del ciclo while l < Longitud(frase); regresa al do
 - g. Imprimir palabraMayor, mayorTamaño
 - h. Fin Función principal
2. Función obtenerPalabra(Val fra: Cadena; Ref c: Entero) Cadena

Declara fra y c como parámetros que se conectan con frase y l.

 - a. Se declaran variables
 - b. Inicia pal en nulo
 - c. Inicia ciclo do
 1. Copia en car el siguiente carácter
 2. Si car != ' ' Entonces
 - a. Concatena en pal el valor de car

3. Fin if
 4. Incrementa c en 1
 - d. Fin ciclo while (car != ' ')AND(c != Longitud(fra)); Regresa al do
 - e. Regresa el valor de pal
 - f. Fin Función obtenerPalabra
- Fin del algoritmo

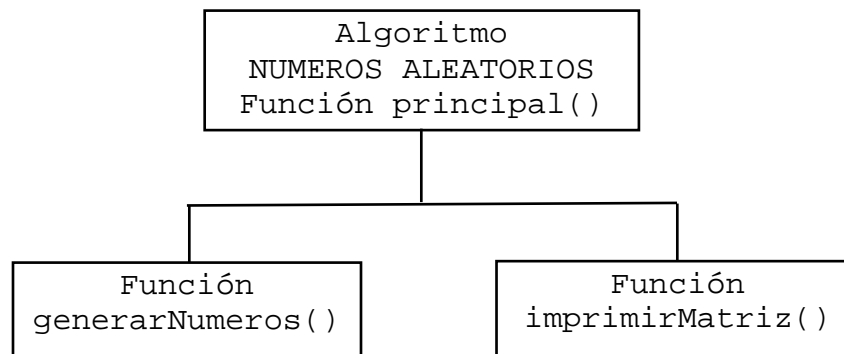
Ejercicio 7.6.18

Elaborar un algoritmo que genere números aleatorios (al azar) y los guarde en una matriz de 20x10 (generará 200 números); y por último que los imprima. Usar un arreglo de dos dimensiones como parámetro.

A continuación se tiene el algoritmo de la solución:

(Primero hágalo usted...después compara la solución)

A continuación se tiene el diagrama general de la solución:



Ahora tenemos el algoritmo en pseudocódigo:

Algoritmo NUMEROS ALEATORIOS

1. Función principal()
 - a. Declarar
 - Variables
 - numeros: Arreglo[20][10] Real
 - b. generarNumeros(numeros)
 - c. imprimirMatriz(numeros)
 - d. Fin Función principal
2. Función generarNumeros(Ref mat: Arreglo[20][10] Real)
 - a. Declarar
 - Variables
 - r, c: Entero
 - b. IniciaRandom
 - c. for r=0; r<=19; r++

```

        1. for c=0; c<=9; c++
            a. mat[r][c] = Random
        2. endfor
    d. endfor
    e. Fin Función generarNumeros

3. Función imprimirMatriz(Val ma: Arreglo[20][10] Real)
    a. Declarar
        Variables
        re, co: Entero
    b. for re=0; re<=19; re++
        1. for co=0; co<=9; co++
            a. Imprimir ma[re][co]
        2. endfor
    c. endfor
    d. Fin Función imprimirMatriz
Fin

```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C727.C
Programa en Java: NumerosAleatorios.java

Explicación:

1. Función principal()
 - a. Se declara la variable numeros que es un arreglo de 20,10
 - b. Llama generarNumeros
 - c. Llama imprimirMatriz
 - d. Fin Función principal

2. Función GenerarNumeros ()
 - a. Se declaran las variables r y c
 - b. Inicia generador de números aleatorios Random
 - c. Inicia ciclo for desde r = 0 hasta 19
 1. Inicia ciclo for desde c = 0 hasta 9
 - a. Genera número Random y lo pone en numeros[r,c]
 2. Fin for
 - d. Fin for
 - e. Fin Función generarNumeros

3. Función imprimirMatriz()
 - a. Se declaran las variables re y co
 - b. Inicia ciclo for desde re = 0 hasta 19
 1. Inicia ciclo for desde co = 0 hasta 9
 - a. Imprime numeros[re,co]
 2. Fin for
 - c. Fin for
 - d. Fin Función imprimirMatriz

Fin del algoritmo