Capítulo 6 WEB

Con formato: Español (alfab. internacional)

6.1.1 Ejercicios resueltos para unidimensionales (Continuación...)

Ejercicio 6.1.1.4

Elaborar un algoritmo que permita leer 15 números en un arreglo de 15 elementos; calcular la media, imprimir cada elemento del arreglo y la desviación que tiene respecto a la media de la siguiente forma:

ELEMENTO DESVIACIÓON

---	---
---	---
---	---
---	---
---	---

MEDIA = ---

Cálculos:
MEDIA se obtiene de la sumatoria de los 15 elementos del arreglo y se divide entre 15.
DESVIACIÓON de cada elemento se obtiene restándole al elemento la MEDIA.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo DESVIACIÓON DE MEDIA
1. Declarar
     Variables
       numeros: Arreglo[15] Entero
       r, sumatoria: Entero
       media, desviacion: Real
2. sumatoria = 0
3. for r=0; r<=14; r++
       a. Solicitar núumero r+1
       b. Leer núumeros[r]
       c. sumatoria = sumatoria + numerosnúmeros[r]
4. endfor
5. media = sumatoria / r
6. Imprimir encabezado
7. for r=0; r<=14; r++
       a. desviaciondesviación = numerosnúmeros[r] - media
       b. Imprimir numerosnúmeros[R], desviaciondesviación
8. endfor
9. Imprimir media
10. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C605.C y
Programa en Java: DesviacionMedia.java

Explicación:
1. Se declaran las variables
2. Inicia sumatoria en 0
3. Inicia ciclo for desde r=0 hasta 14
   a. Solicita nú~~u~~mero r
   b. Se lee en numeros[r]
   c. Incrementa sumatoria con numeros[r]
4. Fin del for
5. Calcula media = sumatoria / r
6 Imprime encabezado
7. Inicia ciclo for desde r=0 hasta 14
   a. Calcula ~~desviacion~~desviación = numeros[r] ~~-~~ media
   b. Imprime numeros[r], ~~desviacion~~desviación
8. Fin del for
9. Imprime media
10. Fin del algoritmo

Ejercicio 6.1.1.5

Elaborar un algoritmo que lea el nombre de un obrero y las unidades producidas por éste los 30 días del mes y que, además, imprima el siguiente reporte:

Nombre del obrero: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
Producción del día 1: 999
Producción del día 2: 999
.
.
.
Producción del día 30: 999

Promedio de producción: 999.99
Total de días arriba del promedio: 99
Cantidad producida más alta: 999
Día más productivo: 99

Cálculos:
- Promedio de producción. Se calcula mediante la suma de todos los días de producción dividido entre 30.
- Total de días arriba del promedio. Es el total de días en que la producción fue mayor que el promedio. Se compara la producción de cada día con el promedio y se cuentan sólo los días cuya producción es mayor que éste.

Con formato: Sangría: Izquierda: 0 cm, Sangría francesa: 0.32 cm

– Cantidad producida más alta. Comparar la producción de todos los días para determinar la que es mayor; se supone que son diferentes.
– Día más productivo. El número de día al que corresponda la cantidad producida más alta.

(Primero hágalo usted, después compare la solución.)

```
Algoritmo PRODUCCIÓN PROMEDIO
1. Declarar
     Variables
       nombreObr: Caden
       produccion: Arreglo[30]  Entero
       i, totArriba, sumaProd, diaMayor,
       prodMayor: Entero
       promedio: Real
2. sumaProd = 0
3. Solicitar nombre del obrero
4. Leer nombreObr
5. for i=0; i<=29; i++
       a. Solicitar producción del día i+1
       b. Leer produccion[i]
       c. sumaProd = sumaProd + produccion[i]
6. endfor
7. promedio = sumaProd / 30
8. totArriba = 0
9. prodMayor = 0
10. Imprimir nombreObr
11. for i=0; i<=29; i++
        a. Imprimir produccion[i]
        b. if produccion[i] > promedio then
              1. totArriba = totArriba + 1
        c. endif
        d. if produccion[i] > prodMayor then
              1. prodMayor = produccion[i]
              2. diaMayor = i
        e. endif
12. endfor
13. Imprimir promedio, totArriba, prodMayor, diaMayor
14. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C606.C y
Programa en Java: ProduccionObrero.java

Explicación:
1. Se declaran las variables
2. Se inicia sumaProd en 0

3. Solicita nombre del obrero
4. Se lee en nombreObr
5. Inicia ciclo for desde i=0 hasta 29 con incrementos de 1
    a. Solicita producción del día i+1
    b. Se lee en produccion[i]
    c. Incrementa sumaProd con produccion[i]
6. Fin del for
7. Calcula promedio = sumaProd / 30
8. Inicia totArriba en 0
9. Inicia prodMayor en 0
10. Imprime nombreObr
11. Inicia ciclo for desde i=0 hasta 29 con incrementos de 1
    a. Imprime produccion[i]
    b. Si produccion[i] > promedio entonces
        1. Incrementa totArriba en 1
    c. Fin del if
    d. Si produccion[i] > prodMayor entonces
        1. Coloca produccion[i] en prodMayor
        2. Coloca i en diaMayor
    e. Fin del if
12. Fin del for
13. Imprime promedio, totArriba, prodMayor, diaMayor
14. Fin del algoritmo

Ejercicio 6.1.1.6

Elaborar un algoritmo que permita leer 10 números en un arreglo. A continuación preguntar si desea introducir un nuevo valor, si es así, debe leer el nuevo valor y meterlo en la posición 0 del arreglo y todos los demás recorrerlos a la siguiente posición. El elemento de la posición 9 se perderá, es decir, saldrá del arreglo, porque el valor que se introduce "empuja" a los demás. Enseguida debe imprimirse todo el arreglo. Esto deberá repetirse mientras desee introducir un nuevo valor.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo FILA DE DATOS
1. Declarar
     Variables
       fila: Arreglo [10] Entero
       i, valor: Entero
       desea: Carácter
2. for i=0; i<=9; i++
       a. Solicitar elemento i de la fila
       b. Leer fila[i]
3. endfor
4. Preguntar "¿Desea introducir valor (S/N)?"
```

```
5. Leer desea
6. while desea == 'S'
          a. Se solicita el valor a introducir
          b. Leer valor
          c. for i=9; i>=1; i--
               1. fila[i+1] = fila[i]
          d. endfor
          e. fila[0] = valor
          f. for i=0; i<=9; i++
               1. Imprimir fila[i]
          g. endfor
          h. Preguntar "¿Desea introducir el valor (S/N)?"
          i. Leer desea
7. endwhile
8. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C607.C y
Programa en Java: Fila.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde i=0 hasta 9 con incrementos de 1
   a. Solicita elemento i de la fila
   b. Se lee en fila[i]
3. Fin del for
4. Pregunta "¿Desea introducir valor (S/N)?"
5. Lee en desea la respuesta
6. Inicia ciclo while Si desea=='S' entonces
   a. Se solicita el valor a introducir
   b. Se lee en valor
   c. Inicia ciclo for desde i=9 hasta 1 con decrementos de -1
      1. Coloca fila[i] en fila[i+1]
   d. Fin del for
   e. Coloca valor en fila[0]
   f. Inicia ciclo for desde i=0 hasta 9 con incrementos de 1
      1. Imprime fila[i]
   g. Fin del for
   h. Pregunta "¿Desea introducir el valor (S/N)?"
   i. Se lee en desea la respuesta
7. Fin del while
8. Fin del algoritmo

Con formato: Color de fuente: Rojo

Ejercicio 6.1.1.7

Elaborar un algoritmo que lea 20 números enteros en un arreglo, que imprima el arreglo, el número mayor y cuántos elementos hay de ~~é~~este número.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo MAYOR 20 NÚMEROS
1. Declarar
     Variables
       numeros: Arreglo[20] Entero
       r, mayor, totIguales: Entero
2. for r=0; r<=19; r++
       a. Solicitar número r
       b. Leer numeros[r]
3. endfor
4. for r=0; r<=19; r++
       a. Imprimir numeros[r]
5. endfor
6. mayor = numeros[0]
7. totIguales = 1
8. for r=1; r<=19; r++
       a. if numeros[r] > mayor then
             1. mayor = numeros[r]
             2. totIguales = 1
       b. else
             1. if numeros[r] == mayor then
                   a. totIguales = totIguales + 1
             2. endif
       c. endif
9. endfor
10. Imprimir mayor, totIguales
11. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C608.C y
Programa en Java: mayor20numeros.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 19
   a. Solicita número r
   b. Se lee en numeros[r]
3. Fin del for
4. Inicia ciclo for desde r=0 hasta 19
   a. Imprime numeros[r]
5. Fin del for
6. Coloca numeros[0] en mayor
7. Inicia totIguales en 1
8. Inicia ciclo for desde r=1 hasta 19
   a. Si numeros[r] > mayor entonces

1. Coloca numeros[r] en mayor
2. Inicia totIguales en 1

b. Si no
1. Si numeros[r] == mayor entonces
a. Incrementa totIguales en 1
2. Fin del if

c. Fin del if

9. Fin del for
10. Imprime mayor, totIguales
11. Fin del algoritmo

Ejercicio 6.1.1.8

Elaborar un algoritmo que permita leer los nombres de 10 personas en un arreglo de una dimensión, los pesos de esas 10 personas en otro arreglo y las estaturas de las mismas en otro arreglo de una dimensión, es decir, se tendrán un arreglo de nombres, otro de pesos y otro de estaturas; imprimir el siguiente reporte:

REPORTE DE PERSONAS

NOMBRE	PESO	ESTATURA
XXXXXXXXXXXXXXXXXXXX	99.9	99.9
XXXXXXXXXXXXXXXXXXXX	99.9	99.9
.		
.		
XXXXXXXXXXXXXXXXXXXX	99.9	99.9

PROMEDIO PESO: 99.9
PROMEDIO ESTATURA: 99.9

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo PESOS Y ESTATURAS
1. Declarar
     Variables
       nombres: Arreglo[10] Cadena
       pesos: Arreglo[10] Real
       estaturas : Arreglo[10] Real
       i: Entero
       totPeso,totEstatura,promPeso,promEstatura: Real
2. for i=0; i<=9; i++
       a. Solicitar nombres[i], pesos[i], estaturas[i]
       b. Leer nombres[i], pesos[i], estaturas[i]
3. endfor
4. Imprimir encabezado
5. totPeso = 0 ; totEstatura = 0
6. for i=0; i<=9; i++
```

```
      a. Imprimir nombres[i], pesos[i], estaturas[i]
      b. totPeso = totPeso + pesos[i]
         totEstatura = totEstatura + estaturas[i]
7. endfor
8. promPeso = totPeso / 10
   promEstatura = totEstatura / 10
9. Imprimir promPeso, promEstatura
10. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C609.C y
Programa en Java: PesosEstaturas.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde i=0 hasta 9
   a. Solicita nombres[i], pesos[i], estaturas[i]
   b. Se leen en nombres[i], pesos[i], estaturas[i]
3. Fin del for
4. Imprime encabezado
5. Inicia totPeso en 0; totEstatura en 0
6. Inicia ciclo for desde i=0 hasta 9
   a. Imprime nombres[i], pesos[i], Estaturas[i]
   b. Incrementa totPeso con pesos[i]
      Incrementa totEstatura con estaturas[i]
7. Fin del for
8. Calcula promPeso = totPeso / 10
   Calcula promEstatura = totEstatura / 10
9. Imprime promPeso, promEstatura
10. Fin del algoritmo

Ejercicio 6.1.1.9

Elaborar un algoritmo que permita leer los nombres de 15 personas en un arreglo de una dimensión, las edades de esas 15 personas en otro arreglo y los sueldos de las mismas en otro arreglo de una dimensión, es decir, se tendrán un arreglo de nombres, otro de edades y otro de sueldos; imprimir el siguiente reporte:

```
                REPORTE DE EMPLEADOS
NOMBRE                          EDAD        SUELDO
_____________________________________________________
XXXXXXXXXXXXXXXXXXXX      999         99999.99
XXXXXXXXXXXXXXXXXXXX      999         99999.99
        .
        .
XXXXXXXXXXXXXXXXXXXX      999         99999.99
```

TOTAL 999999.99
EMPLEADO CON MAYOR SUELDO: XXXXXXXXXXXXXXXXXXXXXXXXX
SUELDO DEL EMPLEADO QUE GANA ~~MAS~~MÁS: 99999.99
EDAD DEL EMPLEADO QUE GANA MÁ~~A~~S: 999

(Primero hágalo usted, ~~...~~después compare la solución.)

```
Algoritmo EDADES Y SUELDOS
1. Declarar
     Variables
       nombres: Arreglo[15] Cadena
       edades: Arreglo[15] Entero
       sueldos: Arreglo[15] Real
       i, edadMayor: Entero
       sueldoMayor, totSueldos : Real
       empleadoMayor: Cadena
2. for i=0; i<=14; i++
       a. Solicitar nombres[i], edades[i], sueldos[i]
       b. Leer nombres[i], edades[i], sueldos[i]
3. endfor
4. Imprimir encabezado
5. sueldoMayor = 0 ; totSueldos = 0
6. for i=0; i<=14; i++
       a. Imprimir nombres[i], edades[i], sueldos[i]
       b. totSueldos = totSueldos + sueldos[i]
       c. if sueldos[i] > sueldoMayor then
             1. sueldoMayor = sueldos[i]
             2. empleadoMayor = nombres[i]
             3. edadMayor = edades[i]
       d. endif
7. endfor
8. Imprimir totSueldos, empleadoMayor,
           sueldoMayor, edadMayor
9. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C610.C y
Programa en Java: EdadesSueldos.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde i=0 hasta 14
   a. Solicita nombres[i], edades[i], sueldos[i]
   b. Se leen en nombres[i], edades[i], sueldos[i]
3. Fin del for
4. Imprime encabezado
5. Inicia sueldoMayor en 0; totSueldos en 0

6. Inicia ciclo for desde i=0 hasta 14
   a. Imprime nombres[i], edades[i], sueldos[i]
   b. Incrementa totSueldos con sueldos[i]
   c. Si sueldos[i] > sueldoMayor entonces
      1. Coloca sueldos[i] en sueldoMayor
      2. Coloca nombres[i] en empleadoMayor
      3. Coloca edades[i] en edadMayor
   d. Fin del if
7. Fin del for
8. Imprime totSueldos, empleadoMayor, sueldoMayor, edadMayor
9. Fin del algoritmo

6.2.1 Ejercicios resueltos para bidimensionales (Continuación...)

Ejercicio 6.2.1.4

Elaborar un algoritmo que lea números en una matriz de 4X5 e imprima ésta y la transpuesta. La transpuesta de una matriz de orden mxn, es una matriz de orden nxm que se obtiene intercambiando filas por columnas, es decir, el elemento Aij, se coloca en el Bji.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo TRANSPUESTA
1. Declarar
     Variables
       a: Arreglo[4][5] Entero
       b: Arreglo[5][4] Entero
       r, c: Entero
2. for r=0; r<=3; r++
       a. for c=0; c<=4; c++
              1. Solicitar Elemento r,c de la matriz a
              2. Leer a[r][c]
       b. endfor
3. endfor
4. for r=0; r<=3; r++
       a. for c=0; c<=4; c++
              1. b[c][r]= a[r][c]
       b. endfor
5. endfor
6. for r=0; r<=3; r++
       a. for c=0; c<=4; c++
              1. Imprimir a[r][c]
       b. endfor
7. endfor
8. for r=0; r<=4; r++
```

```
        a. for c=0; c<=3; c++
                1. Imprimir b[r][c]
        b. endfor
9. endfor
10. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C615.C y
Programa en Java: Transpuesta.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 3
    a. Inicia ciclo for desde c=0 hasta 4
        1. Solicita Elemento r,c de la matriz a
        2. Se lee en a[r][c]
    b. Fin del for
3. Fin del for
4. Inicia ciclo for desde r=0 hasta 3
    a. Inicia ciclo for desde c=0 hasta 4
        1. Coloca a[r][c] en b[c][r]
    b. Fin del for
5. Fin del for
6. Inicia ciclo for desde r=0 hasta 3
    a. Inicia ciclo for desde c=0 hasta 4
        1. Imprime a[r][c]
    b. Fin del for
7. Fin del for
8. Inicia ciclo for desde r=0 hasta 4
    a. Inicia ciclo for desde c=0 hasta 3
        1. Imprime b[r][c]
    b. Fin del for
9. Fin del for
10. Fin del algoritmo

Ejercicio 6.2.1.5

Elaborar un algoritmo que permita leer números en una matriz A de 3X5, lo propio para una matriz B, que las imprima e indique si las matrices son iguales o no. Dos matrices son iguales si todos sus elementos correspondientes son iguales.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo IGUALDAD DE MATRICES
1. Declarar
     Variables
       a, b: Arreglo[3][5] Entero
```

```
        r, c, bandera: Entero
2. for r=0; r<=2; r++
        a. for c=0; c<=4; c++
                1. Solicitar Elemento r,c de la matriz a
                2. Leer a[r][c]
         b. endfor
3. endfor
4. for r=0; r<=2; r++
        a. for c=0; c<=4; c++
                1. Solicitar Elemento r,c de la matriz b
                2. Leer b[r][c]
        b. endfor
5. endfor
6. for r=0; r<=2; r++
        a. for c=0; c<=4; c++
                1. Imprimir a[r][c]
        b. endfor
7. endfor
8. bandera = 0
9. for r=0; r<=2; r++
        a. for c=0; c<=4; c++
                1. Imprimir b[r][c]
                2. if a[r][c] != b[r][c] then
                       a. bandera = 1
                3. endif
        b. endfor
10. endfor
11. if bandera == 0 then
        a. Imprimir "SON IGUALES"
12. else
        a. Imprimir "NO SON IGUALES"
13. endif
14. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C616.C y
Programa en Java: IgualdadMatrices.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 2
   a. Inicia ciclo for desde c=0 hasta 4
      1. Solicita Elemento r,c de la matriz a
      2. Se lee en a[r][c]
   b. Fin del for
3. Fin del for
4. Inicia ciclo for desde r=0 hasta 2

a. Inicia ciclo for desde c=0 hasta 4
1. Solicita Elemento r,c de la matriz b
2. Se lee en b[r][c]
b. Fin del for
5. Fin del for
6. Inicia ciclo for desde r=0 hasta 2
a. Inicia ciclo for desde c=0 hasta 4
1. Imprime a[r][c]
b. Fin del for
7. Fin del for
8. Inicia bandera en 0
9. Inicia ciclo for desde r=0 hasta 2
a. Inicia ciclo for desde c=0 hasta 4
1. Imprime b[r][c]
2. Si a[r][c] != b[r][c] entonces
a. Coloca 1 en bandera, para ~~inducar~~indicar que no hay igualdad
3. Fin del if
b. Fin del for
10. Fin del for
11. Si bandera==0 entonces
a. Imprime "SON IGUALES"
12. Si no
a. Imprime "NO SON IGUALES"
13. Fin del if
14. Fin del algoritmo

Ejercicio 6.2.1.6

Elaborar un algoritmo que permita leer números en una matriz A de 5X6, lo propio para un vector X de 5 elementos. Calcular el producto de matriz por vector columna: el producto de la matriz A por el vector X, es un vector Z donde el elemento 1 está~~a~~ dado por la sumatoria de los productos del elemento 1 del vector X por cada uno de los elementos del renglón 1 de la matriz A, lo propio para el 2, 3, etc. Imprimir la matriz A, el vector X y el vector Z.

(Primero hágalo usted, ~~...~~después compare la solución.)

```
Algoritmo MATRIZ POR VECTOR COLUMNA
1. Declarar
     Variables
       a: Arreglo[5][6] Entero
       x, z: Arreglo[5] Entero
       r, c: Entero
2. for r=0; r<=4; r++
       a. for c=0; c<=5; c++
              1. Solicitar Elemento r,c de la matriz a
              2. Leer a[r][c]
```

```
        b. endfor
3. endfor
4. for r=0; r<=4; r++
        a. Solicitar Elemento r del vector x
        b. Leer x[r]
5. endfor
6. for r=0; r<=4; r++
        a. z[r] = 0
        b. for c=0; c<=5; c++
                1. z[r] = z[r] + (a[r][c]* x[r])
        ca. endfor
7. endfor
8. for r=0; r<=4; r++
        a. for c=0; c<=5; c++
                1. Imprimir a[r][c]
        b. endfor
        c. Imprimir x[r], z[r]
9. endfor
10. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C617.C y
Programa en Java: MatrizVector.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 4
    a. Inicia ciclo for desde c=0 hasta 5
        1. Solicita Elemento r,c de la matriz a
        2. Se lee en a[r][c]
    b. Fin del for
3. Fin del for
4. Inicia ciclo for desde r=0 hasta 4
    a. Solicita Elemento r del vector x
    b. Se lee en x[r]
5. Fin del for
6. Inicia ciclo for desde r=0 hasta 4
    a. Inicia z[r] en 0
    b. Inicia ciclo for desde c=0 hasta 5
        1. Calcula z[r] = z[r] + (a[r][c]* x[r])
    c. Fin del for
7. Fin del for
8. Inicia ciclo for desde r=0 hasta 4
    a. Inicia ciclo for desde c=0 hasta 5
        1. Imprime a[r][c]
    b. Fin del for
    c. Imprime x[r], z[r]

9. Fin del for
10. Fin del algoritmo

Ejercicio 6.2.1.7

Elaborar un algoritmo que permita leer números en una matriz de 4X5, que la imprima, que calcule e imprima la sumatoria por renglones y por columnas, además que imprima el número de renglón y el número de columna que tuvieron la mayor sumatoria.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo SUMAS POR RENGLONES Y COLUMNAS
1. Declarar
     Variables
       matriz: Arreglo[4][5] Entero
       r, c, renMay, colMay, sumaRen, SumaCol,
       mayorRen, mayorCol: Entero
2. for r=0; r<=3; r++
       a. for c=0; c<=4; c++
              1. Solicitar matriz[r][c]
              2. Leer matriz[r][c]
       b. endfor
3. endfor
4. mayorRen = -9999
5. for r=0; r<=3; r++
       a. sumaRen =0
       b. for c=0; c<=4; c++
              1. Imprimir matriz[r][c]
              2. sumaRen = sumaRen + matriz[r][c]
       c. endfor
       d. Imprimir sumaRen
       e. if sumaRen > mayorRen then
             1. mayorRen = sumaRen
             2. renMay = r
       f. endif
6. endfor
7. mayorCol = -9999
8. for c=0; c<=4; c++
       a. sumaCol =0
       b. for r=0; r<=3; r++
              1. sumaCol = sumaCol + matriz[r][c]
       c. endfor
       d. Imprimir sumaCol
       e. if sumaCol > mayorCol then
             1. mayorCol = sumaCol
             2. colMay = c
```

```
        f. endif
9. endfor
10. Imprimir renMay, colMay
11. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C618.C y
Programa en Java: SumasRenCol.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 3
    a. Inicia ciclo for desde c=0 hasta 4
        1. Solicita matriz[r][c]
        2. Se lee en matriz[r][c]
    b. Fin del for
3. Fin del for
4. Inicia mayorRen en --9999
5. Inicia ciclo for desde r=0 hasta 3
    a. Inicia sumaRen en 0
    b. Inicia ciclo for desde c=0 hasta 4
        1. Imprime matriz[r][c]
        2. Incrementa sumaRen con matriz[r][c]
    c. Fin del for
    d. Imprime sumaRen
    e. Si sumaRen > mayorRen entonces
        1. Coloca sumaRen en mayorRen
        2. Coloca r en renMay
    f. Fin del if
6. Fin del for
7. Inicia mayorCol en -9999
8. Inicia ciclo for desde c=0 hasta 4
    a. Inicia sumaCol en 0
    b. Inicia ciclo for desde r=0 hasta 3
        1. Incrementa sumaCol con matriz[r][c]
    c. Fin del for
    d. Imprime sumaCol
    e. Si sumaCol > mayorCol entonces
        1. Coloca sumaCol en mayorCol
        2. Coloca c en colMay
    f. Fin del if
9. Fin del for
10. Imprime renMay, colMay
11. Fin del algoritmo

Ejercicio 6.2.1.8

Una compañía manufacturera fabrica 10 artículos diferentes y se trabajan tres turnos.
Elaborar un algoritmo que permita leer el nombre de cada artículo y la producción que se hizo en cada uno de los tres turnos del día; utilizar un arreglo de una dimensión para leer los nombres de los artículos y un arreglo de dos dimensiones (10X3) para leer la producción de los diez artículos (uno en cada renglón) en los tres turnos una columna para cada turno. La idea es leer el nombre del primer artículo y luego la producción hecha en cada uno de los tres turnos, luego procesar el artículo 2, posteriormente el 3 y así sucesivamente. Imprimir el siguiente reporte:

REPORTE DIARIO DE PRODUCCCIÓN

ARTÍCULO	TURNO 1	TURNO 2	TURNO 3	TOT. PROD.
XXXXXXXXXXXXXXX	999	999	999	999
XXXXXXXXXXXXXXX	999	999	999	999
.				
.				
.				
XXXXXXXXXXXXXXX	999	999	999	999
TOTAL	999	999	999	999

ARTÍCULO CON MAYOR PRODUCCIÓN: XXXXXXXXXXXXXXXXXXXXX
PRODUCCIÓN DEL ARTÍCULO MAYOR: 999

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo PRODUCCIÓN TURNOS
1. Declarar
     Variables
       articulos: Arreglo[10] Cadena
       prod: Arreglo[10][3] Entero
       r,c,mayorProd,totTurno,totProd,toTotProd: Entero
       articuloMay: Cadena
2. for r=0; r<=9; r++
       a. Solicitar articulos[r]
       b. Leer articulos[r]
       c. for c=0; c<=2; c++
              1. Solicitar prod[r][c]
              2. Leer prod[r][c]
      d. endfor
3. endfor
4. Imprimir encabezado
5. toTotProd = 0
   mayorProd =0
6. for r=0; r<=9; r++
```

```
        a. Imprimir articulos[r]
        b. totProd =0
        c. for c=0; c<=2; c++
                1. Imprimir prod[r][c]
                2. totProd = totProd + prod[r][c]
        d. endfor
        e. Imprimir totProd
        f. if totProd > mayorProd then
              1. mayorProd = totProd
              2. articuloMay = articulos[r]
        g. endif
        h. toTotProd = toTotProd + totProd
7. endfor
8. for c=0; c<=2; c++
        a. totTurno = 0
        b. for r=0; r<=9; r++
                1. totTurno = totTurno + prod[r][c]
        c. endfor
        d. Imprimir totTurno
9. endfor
10. Imprimir toTotProd, articuloMay, mayorProd
11. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C619.C y
Programa en Java: ProduccionTurnos.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 9
    a. Solicita articulos[r]
    b. Se lee en articulos[r]
    c. Inicia ciclo for desde c=0 hasta 2
        1. Solicita prod[r][c]
        2. Se lee en prod[r][c]
    d. Fin del for
3. Fin del for
4. Imprimir encabezado
5. Inicia totProd y mayorProd en 0
6. Inicia ciclo for desde r=0 hasta 9
    a. Imprime articulos[r]
    b. Inicia proDia en 0
    c. Inicia ciclo for desde c=0 hasta 2
        1. Imprim prod[r][c]
        2. Incrementa proDia con prod[r][c]
    d. Fin del for
    e. Imprime proDia

f. Si proDia>mayorProd entonces
1. Coloca proDia en mayorProd
2. Coloca articulos[r] en articuloMay
g. Fin del if
h. Incrementa totProd con proDia
7. Fin del for
8. Inicia ciclo for desde c=0 hasta 2
a. Inicia prodTurno en 0
b. Inicia ciclo for desde r=0 hasta 9
1. Incrementa prodTurno con prod[r][c]
c. Fin del for
d. Imprime prodTurno
9. Fin del for
10. Imprime totProd, articuloMay, mayorProd
11. Fin del algoritmo

Ejercicio 6.2.1.9

Elaborar un algoritmo que permita leer el nombre de los 32 estados del país y sus respectivas capitales en un arreglo de 32X2, donde cada renglón se utilice para un estado; en la columna 1 se coloca el nombre del estado y en la columna 2 el nombre de la capital. A continuación, debe permitir realizar consultas (mientras se desee), se introduce el nombre del estado y se deberá imprimir la capital del mismo, es decir, se busca el estado en la columna 1 y donde se encuentre se imprime la columna 2.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo CAPITALES ESTADOS
1. Declarar
     Variables
       estados: Arreglo[32][2] Cadena
       estado: Cadena
       r: Entero
       desea: Carácter
2. for r=0; r<=31; r++
       a. Solicitar estado, r
       b. Leer estados[r][0]
       c. Solicitar capital, r
       d. Leer estados[r][1]
3. endfor
4. Preguntar "¿Desea consultar capital (S/N)?"
5. Leer desea
6. while desea == 'S'
      a. Solicitar estado
      b. Leer estado
      c. r = -1
```

```
      d. DO
            a. r = r + 1          %favor de verificar%
            b. if estado == estados[r][0] then
                  1. Imprimir estados[r][1]
            c. endif
      e. while estado != estados[r][0]
      f. Preguntar "¿Desea consultar capital (S/N)?"
      g. Leer desea
7. endwhile
8. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C620.C y
Programa en Java: Capitales.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 31
   a. Solicita estado, r
   b. Se lee en estados[r][0]
   c. Solicita capital, r
   d. Se lee en estados[r][1]
3. Fin del for
4. Pregunta "¿Desea consultar capital (S/N)?"
5. Lee en desea la respuesta
6. Inicia ciclo while Si desea == 'S'
   a. Solicitar estado a consultar
   b. Se lee en estado
   c. Inicia r en -1
   d. Inicia ciclo do
      1. Incrementa r en 1
      2. Si estado == estados[r][0] entonces
         a. Imprime estados[r][1]
      3. Fin del if
   fe. Fin ciclo do mientras que estado!=estados[r][0]
   fg. Pregunta "¿Desea consultar capital (S/N)?"
   gh. Lee en desea
7. Fin del while
8. Fin del algoritmo

Ejercicio 6.2.1.10

Elaborar un algoritmo que permita leer 15 rangos en un arreglo de 15X3, cada renglón se utilizará para un rango de la siguiente forma: la columna 1 se utilizará para leer el límite inferior del rango, la columna 2 para el límite superior y la columna 3 se utilizará como un contador, el cual se iniciará en cero en el momento de hacer la lectura de cada rango. A continuación se deben leer 50 números, cada

Con formato: Color de fuente: Rojo
Con formato: Color de fuente: Rojo
Con formato: Color de fuente: Rojo
Con formato: Color de fuente: Rojo
Con formato: Color de fuente: Rojo

número debe ser mayor o igual al límite inferior del primer rango y menor o igual al límite superior del último rango. Cada número leído debe buscarse en cuál rango está, se trata de contar cuántos números cayeron en cada rango, en otras palabras, hacer una distribución de frecuencias. Se debe imprimir el siguiente reporte:

```
                    DISTRIBUCIÓON DE FRECUENCIAS
 RANGO      LÍMITE INFERIOR      LÍMITE SUPERIOR      No. OCURRENCIAS
_____________________________________________________________________

   1            999                  999                  999
   2            999                  999                  999
   .
   .

  15            999                  999                  999
```

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo DISTRIBUCIÓON DE FRECUENCIAS
1. Declarar
     Variables
       rangos: Arreglo[15][3] Entero
       i, c, r, numero: Entero
2. for r=0; r<=14; r++
       a. Solicitar LÍMITE INFERIOR rango, r
       b. Leer rangos[r][0]
       c. Solicitar LÍMITE SUPERIOR rango, r
       d. Leer rangos[r][1]
       e. rangos[r][2] = 0
3. endfor
4. for c=1; c<=50; c++
       a. Solicitar número
       b. Leer número
       c. r = -1
       d. do
            1. r=r+1
            2. if (numero>=rangos[r][0])AND
                      (numero<=rangos[r][1]) then
                  a. rangos[r][2] = rangos[r][2]+1
            3. endif
       e. while NOT((numero>=rangos[r][0])AND
                                    (numero<=rangos[r][1]))
5. endfor
6. Imprimir encabezado
7. for r=0; r<=14; r++
       a. Imprimir r
```

```
        b. for c=0; c<=2; c=c+1
                1. Imprimir rangos[r][c]
        c. endfor
8. endfor
9. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C621.C y
Programa en Java: Distribucion.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 14
   a. Solicita LÍMITE INFERIOR rango, r
   b. Se lee en rangos[r][0]
   c. Solicita LÍMITE SUPERIOR rango, r
   d. Se lee en rangos[r][1]
   e. Inicia rangos[r][2] en 0
3. Fin del for
4. Inicia ciclo for desde c=1 hasta 50
   a. Solicita ~~numero~~número
   b. Se lee r en ~~numero~~número
   c. Inicia r en ~~-~~−1
   d. Inicia ciclo do
      1. Incrementa r en 1
      2. Si (numero>=rangos[r][0])AND(numero<=rangos[r][1]) entonces
         a. Incrementa rangos[r][2] en 1
      3. Fin del if
   e. Fin ciclo do mientras NOT((numero>=rangos[r][0])AND
      (numero<=rangos[r][1]))
5. Fin del for
6. Imprime encabezado
7. Inicia ciclo for desde r=0 hasta 14
   a. Imprime r
   b. Inicia ciclo for desde c=0 hasta 2
      1. Imprime rangos[r][c]
   c. Fin del for
8. Fin del for
9. Fin del algoritmo

Ejercicio 6.2.1.11

Elaborar un algoritmo que permita leer números en un arreglo de 3X3 e indique si es un cuadrado mágico. Es un cuadrado mágico si la sumatoria de los elementos es 15 por renglones, columnas y diagonales.

(Primero hágalo usted, ~~...~~después compare la solución.)

```
Algoritmo CUADRADO MÁAGICO
1. Declarar
     Variables
       cuadrado: Arreglo[3][3] Entero
       r, c,sumDiag1,sumDiag2,sumRen,sumCol,bandera: Entero
2. for r=0; r<=2; r++
       a. for c=0; c<=2; c++
              1. Solicitar Elemento r,c
              2. Leer cuadrado[r][c]
       b. endfor
3. endfor
4. bandera = 0; sumDiag1 = 0
5. for r=0; r<=2; r++
       a. sumRen = 0
       b. for c=0; c<=2; c++
              1. Imprimir cuadrado[r][c]
              2. sumRen = sumRen + cuadrado[r][c]
              3. if r==c then
                     a. sumDiag1 = sumDiag1 + cuadrado[r][c]
              4. endif
       c. endfor
       d. Imprimir sumRen
       e. if sumRen!=15 then
             1. bandera = 1
       f. endif
6. endfor
7. for r=0; r<=2; r++
       a. sumCol = 0
       b. for c=0; c<=2; c++
              1. sumCol = sumCol + cuadrado[r][c]
       c. endfor
       d. Imprimir sumCol
       e. if sumCol!=15 then
             1. bandera = 1
       f. endif
8. endfor
9. sumDiag2 = 0
10. for r=0; r<=2; r++
       a. sumDiag2 = sumDiag2 + cuadrado[r][2-(r-1)]
11. endfor
12. Imprimir sumDiag1, sumDiag2
13. if (sumDiag1!=15)OR(sumDiag2!=15) then
       a. bandera = 1
14. endif
15. if bandera == 0 then
       a. Imprimir "ES UN CUADRADO MÁAGICO"
16. else
       a. Imprimir "NO ES UN CUADRADO MÁAGICO"
17. endif
18. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C622.C y
Programa en Java: CuadradoMagico.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 2
    a. Inicia ciclo for desde c=0 hasta 2
        1. Solicita Elemento r,c
        2. Se lee en cuadrado[r][c]
    b. Fin del for
3. Fin del for
4. Inicia bandera en 0; sumDiag1 en 0
5. Inicia ciclo for desde r=0 hasta 2
    a. Inicia sumRen en 0
    b. Inicia ciclo for desde c=0 hasta 2
        1. Imprime cuadrado[r][c]
        2. Incrementa sumR en cuadrado[r][c]
        3. Si r==c entonces
            1. Incrementa sumDiag1 con cuadrado[r][c]
        4. Fin del if
    c. Fin del for
    d. Imprime sumRen
    e. Si sumRen!=15 entonces
        1. Coloca 1 en bandera
    f. Fin del if
6. Fin del for
7. Inicia ciclo for desde r=0 hasta 2
    a. Inicia sumCol en 0
    b. Inicia ciclo for desde c=0 hasta 2
        1. Incrementa sumCol con cuadrado[r][c]
    c. Fin del for
    d. Imprime sumCol
    e. Si sumCol!=15 entonces
        a. Coloca 1 en bandera
    f. Fin del if
8. Fin del for
9. Inicia sumDiag2 en 0
10. Inicia ciclo for desde r=0 hasta 2
    a. Incrementa sumDiag2 con cuadrado[r][2--(r--1)]
11. Fin del for
12. Imprime sumDiag1, sumDiag2
13. Si (sumDiag1!=15)OR(sumDiag2!=15) entonces
    a. Coloca 1 en bandera
14. Fin del if
15. Si bandera == 0 entonces

a. Imprime “ES UN CUADRADO MÁAGICO”
16. Si no
a. Imprime “NO ES UN CUADRADO MÁAGICO”
17. Fin del if
18. Fin del algoritmo

Ejercicio 6.2.1.12

Dos matrices A y B se pueden multiplicar si el número de columnas de A es igual al número de renglones de B. En otras palabras, que los vectores renglón de A, contengan el mismo número de elementos que los vectores columna de B. La multiplicación de matrices se hace multiplicando cada vector renglón de A por cada uno de los vectores columna de B. Consideremos el siguiente ejemplo:

Se tiene la matriz A de 2X3, con los siguientes valores:

$$A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & 3 & 4 \end{bmatrix}$$

y la matriz B de 3X4, con los valores:

$$B = \begin{bmatrix} 5 & 1 & 3 & 4 \\ 2 & 4 & 6 & 3 \\ 3 & 2 & 1 & 5 \end{bmatrix}$$

Al multiplicar A por B, se obtiene una matriz P de 2X4, es decir, con el número de renglones de A y el número de columnas de B. Cada elemento de P se obtiene

mediante el producto de vector renglón de A por vector columna de B. El proceso en detalle para calcular cada uno de los elementos de P, es el siguiente:

$$P_{11} = [\,2,4,5\,] \times \begin{bmatrix} 5 \\ 2 \\ 3 \end{bmatrix} = 2(5) + 4(2) + 5(3) = 10 + 8 + 15 = 33$$

$$P_{12} = [\,2,4,5\,] \times \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} = 2(1) + 4(4) + 5(2) = 2 + 16 + 10 = 28$$

$$P_{13} = [\,2,4,5\,] \times \begin{bmatrix} 3 \\ 6 \\ 1 \end{bmatrix} = 2(3) + 4(6) + 5(1) = 6 + 24 + 5 = 35$$

$$P_{14} = [\,2,4,5\,] \times \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} = 2(4) + 4(3) + 5(5) = 8 + 12 + 25 = 45$$

$$P_{21} = [\,1,3,4\,] \times \begin{bmatrix} 5 \\ 2 \\ 3 \end{bmatrix} = 1(5) + 3(2) + 4(3) = 5 + 6 + 12 = 23$$

$$P_{22} = [\,1,3,4\,] \times \begin{bmatrix} 1 \\ 4 \\ 2 \end{bmatrix} = 1(1) + 3(4) + 4(2) = 1 + 12 + 8 = 21$$

$$P_{23} = [\,1,3,4\,] \times \begin{bmatrix} 3 \\ 6 \\ 1 \end{bmatrix} = 1(3) + 3(6) + 4(1) = 3 + 18 + 4 = 25$$

$$P_{24} = [\,1,3,4\,] \times \begin{bmatrix} 4 \\ 3 \\ 5 \end{bmatrix} = 1(4) + 3(3) + 4(5) = 4 + 9 + 20 = 33$$

La matriz P resultante es:

$$P = \begin{bmatrix} 33 & 28 & 35 & 45 \\ 23 & 21 & 25 & 33 \end{bmatrix}$$

Elaborar un algoritmo que permita leer números enteros para cada uno de los elementos de una matriz A de 2X3, lo mismo para una matriz B de 3X4 y calcular una matriz P de 2X4, multiplicando la matriz A por la matriz B.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo MULTIPLICACIÓON DE MATRICES
1. Declarar
     Variables
       a: Arreglo[2][3] Entero
       b: Arreglo[3][4] Entero
       p: Arreglo[2][4] Entero
       r, c, i: Entero
2. for r=0; r<=1; r++
       a. for c=0; c<=2; c++
              1. Solicitar Elemento a[r][c]
              2. Leer a[r][c]
       b. endfor
3. endfor
4. for r=0; r<=2; r++
       a. for c=0; c<=3; c++
              1. Solicitar Elemento b[r][c]
              2. Leer b[r][c]
       b. endfor
5. endfor
6. for i=0; i<=1; i++
       a. for c=0; c<=3; c++
              1. p[i][c] = 0
              2. for r=0; r<=2; r++
                     1. p[i][c] = p[i][c]+(a[i][r]*b[r][c])
              3. endfor
       b. endfor
7. endfor
8. for r=0; r<=1; r++
       a. for c=0; c<=2; c++
              1. Imprimir a[r][c]
       b. endfor
9. endfor
10. for r=0; r<=2; r++
        a. for c=0; c<=3; c++
               1.  Imprimir b[r][c]
        b. endfor
11. endfor
12. for r=0; r<=1; r++
        a. for c=0; c<=3; c++
               1. Imprimir p[r][c]
        b. endfor
```

```
13. endfor
14. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C623.C y
Programa en Java: MultiplicaMatrices.java

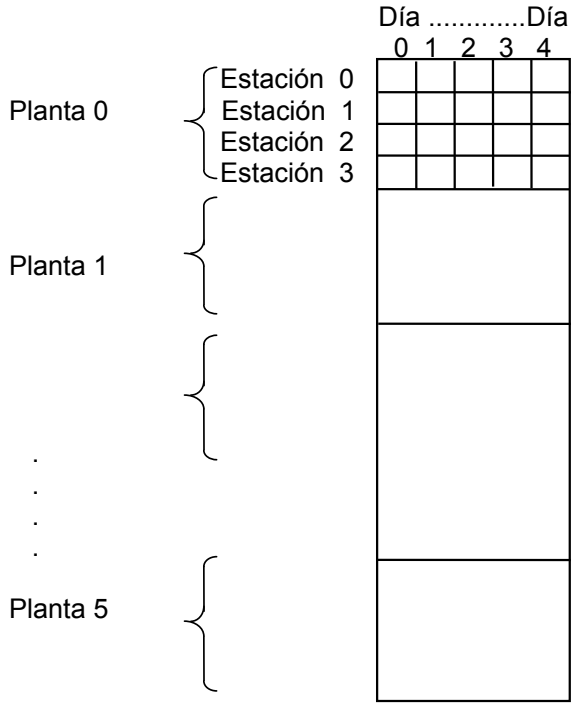
Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde r=0 hasta 1
    a. Inicia ciclo for desde c=0 hasta 2
        1. Solicita Elemento a[r][c]
        2. Se lee en a[r][c]
    b. Fin del for
3. Fin del for
4. Inicia ciclo for desde r=0 hasta 2
    a. Inicia ciclo for desde c=0 hasta 3
        1. Solicita Elemento b[r][c]
        2. Se lee en b[r][c]
    b. Fin del for
5. Fin del for
6. Inicia ciclo for desde i=0 hasta 1
    a. Inicia ciclo for desde c=0 hasta 3
        1. Inicia p[i][c] en 0
        2. Inicia ciclo for desde r=0 hasta 2
            a. Incrementa p[i][c] con (a[i][r] *b[r][c])
        3. Fin del for
    b. Fin del for
7. Fin del for
8. Inicia ciclo for desde r=0 hasta 1
    a. Inicia ciclo for desde c=0 hasta 2
        1. Imprime a[r][c]
    b. Fin del for
9. Fin del for
10. Inicia ciclo for desde r=0 hasta 2
    a. Inicia ciclo for desde c=0 hasta 3
        1. Imprime b[r][c]
    b. Fin del for
11. Fin del for
12. Inicia ciclo for desde r=0 hasta 1
    a. Inicia ciclo for desde c=0 hasta 3
        1. Imprime p[r][c]
    b. Fin del for
13. Fin del for
14. Fin del algoritmo

6.3.1 Ejercicios resueltos para tridimensionales

Ejercicio 6.3.1.1

Una compañía manufacturera tiene 6 plantas, en cada planta tiene 4 estaciones de trabajo y, por cada estación de trabajo se tiene la producción (número de unidades fabricadas) de cada uno de los 5 días laborables de la semana. Elaborar un algoritmo que lea los datos de la producción en un arreglo de tres dimensiones; la primera dimensión la conforman las plantas, la segunda dimensión las estaciones de trabajo, y la tercera, los días de producción, esquemáticamente:



Una vez leídos los datos, que imprima el siguiente reporte:

```
        REPORTE SEMANAL DE PRODUCCIÓN
----------------------  PLANTA 1  ----------------------
                    DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5
                    -------  -------  -------  -------  -------
ESTACIÓN-1          --       --       --       --       --
ESTACIÓN-2          --       --       --       --       --
ESTACIÓN-3          --       --       --       --       --
ESTACIÓN-4          --       --       --       --       --

----------------------  PLANTA 2  ----------------------
```

```
                DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5
                -------  -------  -------  -------  -------
ESTACIÓN-1      --       --       --       --       --
ESTACIÓN-2      --       --       --       --       --
ESTACIÓN-3      --       --       --       --       --
ESTACIÓN-4      --       --       --       --       --
        .
        .
        .
        .
----------------------  PLANTA 6  ----------------------
                DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5
                -------  -------  -------  -------  -------
ESTACIÓN-1      --       --       --       --       --
ESTACIÓN-2      --       --       --       --       --
ESTACIÓN-3      --       --       --       --       --
ESTACIÓN-4      --       --       --       --       --
```

A continuación se presenta el algoritmo:

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO TRES DIMENSIONES
1. Declarar
     Variables
       prod: Arreglo[6][4][5] Entero
       pla, est, dia: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=3; est++
              1. for dia=0; dia<=4; dia++
                     a. Solicitar Producción pla,est,dia
                     b. Leer prod[pla][est][dia]
              2. endfor
       b. endfor
3. endfor
4. for pla=0; pla<=5; pla++
       a. Imprimir encabezado
       b. for est=0; est<=3; est++
              1. for dia=0; dia<=4; dia++
                     a. Imprimir prod[pla][est][dia]
              2. endfor
       c. endfor
5. endfor
6. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C625.C y

Programa en Java: ArregloTresDim2.java

Explicación:

1. Se declaran las variables
   prod es un arreglo de 6 elementos de 4 por 5
   pla, est, dia variables para los ciclos
2. Inicia ciclo for desde pla=0 hasta 5
   a. Inicia ciclo for desde est=0 hasta 3
      1. Inicia ciclo for desde dia=0 hasta 4
         a. Solicita Producción pla,est,dia
         b. Se lee en prod[pla][est][dia]
      2. Fin del for
   b. Fin del for
3. Fin del for
4. Inicia ciclo for desde pla=0 hasta 5
   a. Imprime encabezado
   b. Inicia ciclo for desde est=0 hasta 3
      1. Inicia ciclo for desde dia=0 hasta 4
         a. Imprime prod[pla][est][dia]
      2. Fin del for
   c. Fin del for
5. Fin del for
6. Fin del algoritmo

Ejercicio 6.3.1.2

Elaborar un algoritmo similar al anterior, pero ahora imprimir el total de producción por estación y por cada día para cada planta:

```
          REPORTE SEMANAL DE PRODUCCIÓN
---------------------------  PLANTA 1  ---------------------------
                  DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5  TOTAL
                  -------  -------  -------  -------  -------  ---------
ESTACIÓN-1        --       --       --       --       --       --
ESTACIÓN-2        --       --       --       --       --       --
ESTACIÓN-3        --       --       --       --       --       --
ESTACIÓN-4        --       --       --       --       --       --
                  -------  -------  -------  -------  -------  ---------
   TOTALES        --       --       --       --       --       --
        .
        .
        .
---------------------------  PLANTA 6  ---------------------------
                  DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5  TOTAL
                  -------  -------  -------  -------  -------  ---------
ESTACIOÓN-1       --       --       --       --       --       --
ESTACIÓN-2        --       --       --       --       --       --
```

ESTACIÓN-3 -- -- -- -- -- --
ESTACIÓN-4 -- -- -- -- -- --
------- ------- ------- ------- ------- ---------
TOTALES -- -- -- -- -- --

TOTAL GENERAL DE PRODUCCIÓN --

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO TRES DIMENSIONES
1. Declarar
     Variables
       prod: Arreglo[6][4][5] Entero
       pla, est, dia, totEst, totDia, totPlanta,
       totProd: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=3; est++
              1. for dia=0; dia<=4; dia++
                     a. Solicitar Producción pla,est,dia
                     b. Leer prod[pla][est][dia]
              2. endfor
       b. endfor
3. endfor
4. Imprimir primer renglón del encabezado
5. totProd = 0
6. for pla=0; pla<=5; pla++
       a. Imprimir encabezado
       b. for est=0; est<=3; est++
              1. totEst = 0
              2. for dia=0; dia<=4; dia++
                     a. Imprimir prod[pla][est][dia]
                     b. totEst = totEst + prod[pla][est][dia]
              3. endfor
              4. Imprimir totEst
       c. endfor
       d. totPlanta = 0
       e. for dia=0; dia<=4; dia++
              1. totDia = 0
              2. for est=0; est<=3; est++
                     a. totDia = totDia + prod[pla][est][dia]
              3. endfor
              4. Imprimir totDia
              5. totPlanta = totPlanta + totDia
       f. endfor
       g. Imprimir totPlanta
       h. totProd = totProd + totPlanta
7. endfor
8. Imprimir totProd
9. Fin
```

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

Programa en C: C626.C y
Programa en Java: ArregloTresDim3.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde pla=0 hasta 5
    a. Inicia ciclo for desde est=0 hasta 3
        1. Inicia ciclo for desde dia=0 hasta 4
            a. Solicita Producción pla,est,dia
            b. Se lee en prod[pla][est][dia]
        2. Fin del for
    b. Fin del for
3. Fin del for
4. Imprime primer renglón del encabezado
5. Inicia totProd en 0
6. Inicia ciclo for desde pla=0 hasta 5
    a. Imprime encabezado
    b. Inicia ciclo for desde est=0 hasta 3
        1. Inicia totEst en 0
        2. Inicia ciclo for desde dia=0 hasta 4
            a. Imprime prod[pla][est][dia]
            b. Incrementa totEst con prod[pla][est][dia]
        3. Fin del for
        4. Imprime totEst
    c. Fin del for
    d. Inicia totPlanta en 0
    e. Inicia ciclo for desde dia=0 hasta 4
        1. Inicia totDia en 0
        2. Inicia ciclo for desde est=0 hasta 3
            a. Incrementa totDia con prod[pla][est][dia]
        3. Fin del for
        4. Imprime totDia
        5. Incrementa totPlanta con totDia
    f. Fin del for
    g. Imprime totPlanta
    h. Incrementa totProd con totPlanta
7. Fin del for
8. Imprime totProd
9. Fin del algoritmo

Ejercicio 6.3.1.3

Elaborar un algoritmo similar al anterior, pero ahora imprimir el total de producción por planta, la estación más productiva de la planta y cuánto fue su producción; y el total de producción general, la planta más productiva de la compañía y cuánto fue su producción.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO TRES DIMENSIONES
1. Declarar
     Variables
       prod: Arreglo[6][4][5] Entero
       pla, est, dia, totEst, totDia, totPlanta, totProd,
       mayPla, mayProdPla, mayEst, mayProdEst: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=3; est++
              1. for dia=0; dia<=4; dia++
                     a. Solicitar Producción pla,est,dia
                     b. Leer prod[pla][est][dia]
              2. endfor
       b. endfor
3. endfor
4. Imprimir primer renglón del encabezado
5. totProd = 0; mayProdPla = 0
6. for pla=0; pla<=5; pla++
       a. Imprimir encabezado
       b. mayProdEst = 0
       c. for est=0; est<=3; est++
              1. totEst = 0
              2. for dia=0; dia<=4; dia++
                     a. Imprimir prod[pla][est][dia]
                     b. totEst = totEst + prod[pla][est][dia]
              3. endfor
              4. Imprimir totEst
              5. if totEst > mayProdEst then
                    a. mayProdEst = totEst
                    b. mayEst = est
              6. endif
       d. endfor
       e. totPlanta = 0
       f. for dia=0; dia<=4; dia++
              1. totDia = 0
              2. for est=0; est<=3; est++
                     a. totDia = totDia + prod[pla][est][dia]
              3. endfor
              4. Imprimir totDia
              5. totPlanta = totPlanta + totDia
       g. endfor
       h. Imprimir totPlanta, mayEst, mayProdEst
       i. totProd = totProd + totPlanta
       j. if totPlanta > mayProdPla then
             1. mayProdPla = totPlanta
             2. mayPla = pla
       k. endif
7. endfor
8. Imprimir totProd, mayPla, mayProdPla
9. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C627.C y
Programa en Java: ArregloTresDim4.java

Explicación:

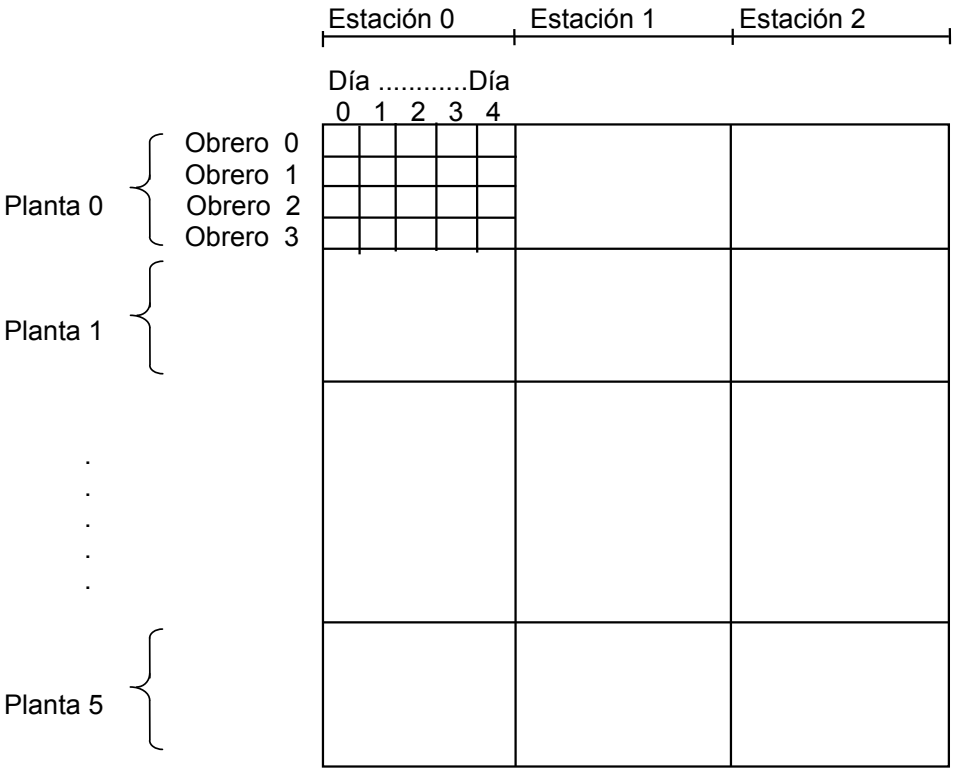
1. Se declaran las variables
2. Inicia ciclo for desde pla=0 hasta 5
    a. Inicia ciclo for desde est=0 hasta 3
        1. Inicia ciclo for desde dia=0 hasta 4
            a. Solicita Producción pla,est,dia
            b. Se lee prod[pla][est][dia]
        2. Fin del for
    b. Fin del for
3. Fin del for
4. Imprime primer renglón del encabezado
5. Inicia totProd en 0; mayProdPla en 0
56. Inicia ciclo for desde pla=0 hasta 5
    a. Imprime encabezado
    b. Inicia mayProdEst en 0
    c. Inicia ciclo for desde est=0 hasta 3
        1. Inicia totEst en 0
        2. Inicia ciclo for desde dia=0 hasta 4
            a. Imprime prod[pla][est][dia]
            b. Incrementa totEst con prod[pla][est][dia]
        3. Fin del for
        4. Imprime totEst
        5. Compara Ssi totEst > mayProdEst, Ssi es asií, entonces
            a. Coloca totEst en mayProdEst
            b. Coloca est en mayEst
        6. Fin del if
    d. Fin del for
    e. Inicia totPlanta en 0
    f. Inicia ciclo for desde dia=0 hasta 4
        1. Inicia totDia en 0
        2. Inicia ciclo for desde est=0 hasta 3
            a. Incrementa totDia con prod[pla][est][dia]
        3. Fin del for
        4. Imprime totDia
        5. Incrementa totPlanta con totDia
    g. Fin del for
    h. Imprime totPlanta, mayEst, mayProdEst
    i. Incrementa totProd con totPlanta
    j. Compara Ssi totPlanta > mayProdPla, -si es asiasí, entonces
        1. Coloca totPlanta -en mayProdPla
        2. Coloca pla en mayPla
    k. Fin del if

7. Fin del for
8. Imprime totProd, mayPla, mayProdPla
9. Fin del algoritmo

6.4.1 Ejercicios resueltos para tetradimensionales

Ejercicio 6.4.1.1

Una compañía manufacturera tiene 6 plantas, en cada planta tiene 3 estaciones de trabajo, en cada estación de trabajo se tienen 4 obreros, y por cada obrero se tiene la producción (número de unidades fabricadas) de cada uno de los 5 días laborables de la semana. Elaborar un algoritmo que lea los datos de la producción en un arreglo de cuatro dimensiones; la primera dimensión la conforman las plantas, la segunda dimensión las estaciones de trabajo, la tercera dimensión los obreros y la cuarta los días de producción, esquemáticamente:



Una vez leídos los datos en el arreglo, imprimir el siguiente reporte:

```
      REPORTE SEMANAL DE PRODUCCIÓON
=====================================
---------------------- PLANTA 1 ----------------------
---------------------- ESTACIÓON 1 ----------------------
```

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5
OBRERO-1	--	--	--	--	--
OBRERO-2	--	--	--	--	--
OBRERO-3	--	--	--	--	--
OBRERO-4	--	--	--	--	--

---------- ESTACIÓN 2 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5
OBRERO-1	--	--	--	--	--
OBRERO-2	--	--	--	--	--
OBRERO-3	--	--	--	--	--
OBRERO-4	--	--	--	--	--

---------- ESTACIÓN 3 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5
OBRERO-1	--	--	--	--	--
OBRERO-2	--	--	--	--	--
OBRERO-3	--	--	--	--	--
OBRERO-4	--	--	--	--	--

==========

.
.

==========

---------- PLANTA 6 ----------

---------- ESTACIÓN 1 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5
OBRERO-1	--	--	--	--	--
OBRERO-2	--	--	--	--	--
OBRERO-3	--	--	--	--	--
OBRERO-4	--	--	--	--	--

---------- ESTACIÓN 2 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5
OBRERO-1	--	--	--	--	--
OBRERO-2	--	--	--	--	--
OBRERO-3	--	--	--	--	--
OBRERO-4	--	--	--	--	--

---------- ESTACIÓN 3 ----------

```
              DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5
             ------- ------- -------  ------- -------
OBRERO-1       --     --     --     --     --
OBRERO-2       --     --     --     --     --
OBRERO-3       --     --     --     --     --
OBRERO-4       --     --     --     --     --
             ------- ------- -------  ------- -------
=====================================
```

A continuación se presenta el algoritmo:

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO TETRADIMENSIONAL
1. Declarar
     Variables
       prod: Arreglo[6][3][4][5] Entero
       pla, est, obr, dia: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=2; est++
              1. for obr=0; obr<=3; obr++
                     a. for dia=0; dia<=4; dia++
                         1. Solicitar
                               producción[pla][est][obr][dia]
                         2. Leer prod[pla][est][obr][dia]
                     b. endfor
              2. endfor
       b. endfor
3. endfor
4. Imprimir primer renglón del encabezado
5. for pla=0; pla<=5; pla++
       a. Imprimir encabezado de planta
       b. for est=0; est<=2; est++
              1. Imprimir encabezado de estación
              2. for obr=0; obr<=3; obr++
                     a. Imprimir "OBRERO-"",Obr
                     b. for dia=0; dia<=4; dia++
                         1. Imprimir prod[pla][est][obr][dia]
                     c. endfor
              3. endfor
       c. endfor
6. endfor
7. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C629.C y
Programa en Java: ArregloCuatroDim2.java

Explicación:

1. Se declaran las variables
   prod es un arreglo de cuatro dimensiones, 6 elementos en la primera dimensión, 3 elementos en la segunda, 4 en la tercera dimensión y 5 elementos en la cuarta.
   pla,est,obr,dia son variables para controlar los cuatro ciclos.
2. Inicia ciclo for desde pla=0 hasta 5
   a. Inicia ciclo for desde est=0 hasta 2
      1. Inicia ciclo for desde obr=0 hasta 3
         a. Inicia ciclo for desde dia=0 hasta 4
            1. Solicita prod[pla][est][obr][dia]
            2. Se lee en prod[pla][est][obr][dia]
         b. Fin del for
      2. Fin del for
   b. Fin del for
3. Fin del for
4. ~~Imprim~~Imprime primer renglón del encabezado
5. Inicia ciclo for desde pla=0 hasta 5
   a. Imprime encabezado de planta
   b. Inicia ciclo for desde est=0 hasta 2
      1. Imprime encabezado de estación
      2. Inicia ciclo for desde obr=0 hasta 3
         a. Imprime "OBRERO-"", obr
         b. Inicia ciclo for desde dia=0 hasta 4
            1. Imprime prod[pla][est][obr][dia]
         c. Fin del for
      3. Fin del for
   c. Fin del for
6. Fin del for
7. Fin del algoritmo

Ejercicio 6.4.1.2

Elaborar un algoritmo similar al anterior, pero ahora imprimir el siguiente reporte:

```
          REPORTE SEMANAL DE PRODUCCIÓ~~O~~N
==========================================
---------------------------  PLANTA 1  ----------------------------
---------------------------- ESTACI~~O~~ÓN 1  ----------------------------
                    DÍ~~I~~A 1  DÍ~~I~~A 2  DÍ~~I~~A 3  DÍ~~I~~A 4  DÍ~~I~~A 5  TOTAL
                    -------  -------  -------  -------  -------  ---------
OBRERO-1              --       --       --       --       --       --
OBRERO-2              --       --       --       --       --       --
OBRERO-3              --       --       --       --       --       --
OBRERO-4              --       --       --       --       --       --
                    -------  -------  -------  -------  -------  ---------
```

Con formato: Espacio Después: 0 pto

Con formato: Español (alfab. internacional)

TOTALES	--	--	--	--	--	--

---------- ESTACIÓN 2 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	TOTAL
OBRERO-1	--	--	--	--	--	--
OBRERO-2	--	--	--	--	--	--
OBRERO-3	--	--	--	--	--	--
OBRERO-4	--	--	--	--	--	--
TOTALES	--	--	--	--	--	--

---------- ESTACIÓN 3 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	TOTAL
OBRERO-1	--	--	--	--	--	--
OBRERO-2	--	--	--	--	--	--
OBRERO-3	--	--	--	--	--	--
OBRERO-4	--	--	--	--	--	--
TOTALES	--	--	--	--	--	--

==========

.

.

==========

---------- PLANTA 6 ----------

---------- ESTACIÓN 1 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	TOTAL
OBRERO-1	--	--	--	--	--	--
OBRERO-2	--	--	--	--	--	--
OBRERO-3	--	--	--	--	--	--
OBRERO-4	--	--	--	--	--	--
TOTALES	--	--	--	--	--	--

---------- ESTACIÓN 2 ----------

	DÍA 1	DÍA 2	DÍA 3	DÍA 4	DÍA 5	TOTAL
OBRERO-1	--	--	--	--	--	--
OBRERO-2	--	--	--	--	--	--
OBRERO-3	--	--	--	--	--	--
OBRERO-4	--	--	--	--	--	--
TOTALES	--	--	--	--	--	--

```
-------------------------------------------------------------------------

---------------------- ESTACIÓN 3 ----------------------
               DÍA 1  DÍA 2  DÍA 3  DÍA 4  DÍA 5  TOTAL
              ------- ------- -------  ------- ------- ---------
OBRERO-1        --      --      --      --      --      --
OBRERO-2        --      --      --      --      --      --
OBRERO-3        --      --      --      --      --      --
OBRERO-4        --      --      --      --      --      --
              ------- ------- -------  ------- -------  ---------
 TOTALES        --      --      --      --      --      --
-------------------------------------------------------------------------
 TOTAL GENERAL DE PRODUCCIÓN                  --
=========================================
```

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO CUATRO DIMENSIONES
1. Declarar
     Variables
       prod: Arreglo[6][3][4][5] Entero
       pla, est, obr, dia, totEst, totDia, totObr,
       totPlanta, totProd: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=2; est++
              1. for obr=0; obr<=3; obr++
                   a. for dia=0; dia<=4; dia++
                         1. Solicitar producción pla,est,obr,dia
                         2. Leer prod[pla][est][obr][dia]
                   b. endfor
              2. endfor
       b. endfor
3. endfor
4. Imprimir primer renglón del encabezado
5. totProd = 0
6. for pla=0; pla<=5; pla++
       a. Imprimir encabezado de planta
       b. totPlanta = 0
       c. for est=0; est<=2; est++
              1. Imprimir encabezado de estación
              2. for obr=0; obr<=3; obr++
                     a. Imprimir 'OBRERO-',Obr
                     b. totObr = 0
                     c. for dia=0; dia<=4; dia++
                           1. Imprimir prod[pla][est][obr][dia]
                           2. totObr = totObr +
                                       prod[pla][est][obr][dia]
                     d. endfor
                     e. Imprimir totObr
              3. endfor
```

```
            4. totEst = 0
            5. for dia=0; dia<=4; dia++
                  a. totDia = 0
                  b. for obr=0; obr<=3; obr++
                        1. totDia = totDia +
                                    prod[pla][est][obr][dia]
                  c. endfor
                  d. Imprimir totDia
                  e. totEst = totEst + totDia
            6. endfor
            7. Imprimir totEst
            8. totPlanta = totPlanta + totEst
      d. endfor
      e. Imprimir totPlanta
      f. totProd = totProd + totPlanta
7. endfor
8. Imprimir TotProd
9. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C630.C y
Programa en Java: ArregloCuatroDim3.java

Explicación:

1. Se declaran las variables
2. Inicia ciclo for desde pla=0 hasta 5
    a. Inicia ciclo for desde est=0 hasta 2
        1. Inicia ciclo for desde obr=0 hasta 3
            a. Inicia ciclo for desde dia=0 hasta 4
                1. Solicita prod[pla][est][obr][dia]
                2. Se lee en prod[pla][est][obr][dia]
            b. Fin del for
        2. Fin del for
    b. Fin del for
3. Fin del for
4. Imprime primer renglón del encabezado
5. Inicia totProd en 0
6. Inicia ciclo for desde pla=0 hasta 5
    a. Imprime encabezado de planta
    b. Inicia totPlanta en 0
    c. Inicia for desde est=0 hasta 2
        1. Imprime encabezado de estación
        2. Inicia ciclo for desde obr=0 hasta 3
            a. Imprime “OBRERO-“,obr
            b. Inicia totObr en 0
            c. Inicia ciclo for desde dia=0 hasta 4
                1. Imprime prod[pla][est][obr][dia]
                2. Incrementa totObr con prod[pla][est][obr][dia]

d. Fin del for
e. Imprime totObr
3. Fin del for
4. Inicia totEst en 0
5. Inicia ciclo for desde dia=0 hasta 4
a. Inicia totDia en 0
b. Inicia ciclo for desde obr=0 hasta 3
1. Incrementa totDia con prod[pla][est][obr][dia]
c. Fin del for
d. Imprime totDia
e. Incrementa totEst con totDia
6. Fin del for
7. Imprime totEst
8. Incrementa totPlanta con totEst
d. Fin del for
e. Imprime totPlanta
f. Incrementa totProd con totPlanta
7. Fin del for
8. Imprime totProd
9. Fin del algoritmo

Ejercicio 6.4.1.3

Elaborar un algoritmo similar al anterior, añadiéndo al final de cada planta imprimir la estación más productiva de la planta y cuánto fué su producción. Y al final del reporte imprimir la planta más productiva de toda la compañía y cuánto fué su producción.

(Primero hágalo usted, ...después compare la solución.)

```
Algoritmo ARREGLO CUATRO DIMENSIONES
1. Declarar
     Variables
       prod: Arreglo[6][3][4][5] Entero
       pla, est, obr, dia, totEst, totDia, totObr,
       totPlanta, totProd, mayEst, mayProdEst,
       mayPla, mayProdPla: Entero
2. for pla=0; pla<=5; pla++
       a. for est=0; est<=2; est++
              1. for obr=0; obr<=3; obr++
                     a. for dia=0; dia<=4; dia++
                         1. Solicitar prod[pla][est][obr][dia]
                         2. Leer prod[pla][est][obr][dia]
                     b. endfor
              2. endfor
       b. endfor
3. endfor
4. Imprimir primer renglón del encabezado
```

```
5. totProd = 0 ; mayProdPla = 0
6. for pla=0; pla<=5; pla++
       a. Imprimir encabezado de planta
       b. totPlanta = 0 ; mayProdEst = 0
       c. for est=0; est<=2; est++
              1. Imprimir encabezado de estación
              2. for obr=0; obr<=3; obr++
                     a. Imprimir 'OBRERO-',obr
                     b. totObr = 0
                     c. for dia=0; dia<=4; dia++
                         1. Imprimir prod[pla][est][obr][dia]
                         2. totObr = totObr +
                                     prod[pla][est][obr][dia]
                     d. endfor
                     e. Imprimir totObr
              3. endfor
              4. totEst = 0
              5. for dia=0; dia<=4; dia++
                     a. totDia = 0
                     b. for obr=0; obr<=3; obr++
                         1. totDia = totDia +
                                     prod[pla][est][obr][dia]
                     c. endfor
                     d. Imprimir totDia
                     e. totEst = totEst + totDia
              6. endfor
              7. Imprimir totEst
              8. totPlanta = totPlanta + totEst
              9. if totEst > mayProdEst then
                     a. mayProdEst = totEst
                     b. mayEst = est
             10. endif
       d. endfor
       e. Imprimir totPlanta, mayEst, mayProdEst
       f. totProd = totProd + totPlanta
       g. if totPlanta > mayProdPla then
             1. mayProdPla = totPlanta
             2. mayPla = Pla
       h. endif
7. endfor
8. Imprimir totProd, mayPla, mayProdPla
9. Fin
```

En la zona de descarga de la Web del libro, están disponibles:
Programa en C: C631.C y
Programa en Java: ArregloCuatroDim4.java

Explicación:
1. Se declaran las variables
2. Inicia ciclo for desde pla=0 hasta 5
   a. Inicia ciclo for desde est=0 hasta 2

1. Inicia ciclo for desde obr=0 hasta 3
    a. Inicia ciclo for desde dia=0 hasta 4
        1. Solicita prod[pla][est][obr][dia]
        2. Se lee en prod[pla][est][obr][dia]
    b. Fin del for
2. Fin del for

b. Fin del for

3. Fin del for
4. Imprime primer renglón del encabezado
5. Inicia totProd en 0; mayProdPla en 0
6. Inicia ciclo for desde pla=0 hasta 5
    a. Imprime encabezado de planta
    b. Inicia totPlanta en 0; mayProdEst en 0
    c. Inicia ciclo for desde est=0 hasta 2
        1. Imprime encabezado de estación
        2. Inicia ciclo for desde obr=0 hasta 3
            a. Imprime “OBRERO-”“,obr
            b. Inicia totObr en 0
            c. Inicia ciclo for desde dia=0 hasta 4
                1. Imprime prod[pla][est][obr][dia]
                2. Incrementa totObr con prod[pla][est][obr][dia]
            d. Fin del for
            e. Imprime totObr
        3. Fin del for
        4. Inicia totEst en 0
        5. Inicia ciclo for desde dia=0 hasta 4
            a. Inicia totDia en 0
            b. Inicia ciclo for desde obr=0 hasta 3
                1. Incrementa totDia con prod[pla][est][obr][dia]
            c. Fin del for
            d. Imprime totDia
            e. Incrementa totEst con totDia
        6. Fin del for
        7. Imprime totEst
        8. Incrementa totPlanta con totEst
        9. Compara Si totEst > mayProdEst, Ssi es asiasí, entonces
            a. Coloca totEst en mayProdEst
            b. Coloca est en mayEst
        10. Fin del if
    d. Fin del for
    e. Imprime totPlanta, mayEst, mayProdEst
    f. Incrementa totProd con totPlanta
    g. Compara Ssi totPlanta > mayProdPla, Ssi es asiasí, entonces
        1. Coloca totPlanta en mayProdPla
        2. Coloca pla en mayPla
    h. Fin del if
7. Fin del for

8. Imprime totProd, mayPla, mayProdPla
9. Fin del algoritmo