

SEBASTIAN DE JESÚS SANTOS VERGARA

1.

Algoritmo puntol

```
// Suponga que n y m son enteros positivos y que dentro de la
//matriz A hay numeros enteros
Dimensionar A[100,100], v[100], u[100]
Definir n, m, ac, k, l Como Entero
Definir prom, dim Como Real
Leer n, m
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        leer A[i,j]
    FinPara
FinPara
ac <- 0
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        ac <- ac + A[i,j]
    FinPara
FinPara
dim <- n*m
prom <- ac/dim
k <- 0
l <- 0
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        Si A[i,j] > prom Entonces
            k <- k+1
            v[k] <- A[i,j]
        SiNo
            Si A[i,j] < prom Entonces
                l <- l+1
                u[l] <- A[i,j]
            FinSi
        FinSi
    FinPara
FinPara
Para i <- 1 Hasta k Hacer
    Escribir v[i]
FinPara
Para i <- 1 hasta l Hacer
    Escribir u[i]
FinPara
FinAlgoritmo
```

2.

Algoritmo punto2

```
// Suponga que n y m son enteros positivos y que dentro de la
//matriz A hay numeros enteros
Dimensionar A[100,100], prom[100]
Definir n, m, ac Como Entero
Leer n, m
Para i <-1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        leer A[i,j]
    FinPara
FinPara
Para i <-1 Hasta n Hacer
    ac <- 0
    Para j <- 1 Hasta m Hacer
        ac <- ac + A[i,j]
    FinPara
    prom[i] <- ac/m
FinPara
Para i <-1 Hasta n Hacer
    Escribir prom[i]
FinPara
```

FinAlgoritmo

3.

Algoritmo punto3

```
// Suponga que n es entero positivo mayor que 3
Dimensionar x[100]
Definir n, i, j, cont, div, l Como Entero
Leer n
i <- 0
j <- 3
Mientras j<n Hacer
    div <-1
    cont <- 0
    Mientras cont<3 y div<=j Hacer
        Si j%div = 0 Entonces
            cont <- cont+1
        FinSi
        div <- div+1
    FinMientras
    Si cont = 2 Entonces
        i = i+1
        x[i] <- j
    FinSi
    j = j+1
FinMientras
Para l <-1 Hasta i Hacer
    Escribir x[l]
FinPara
```

FinAlgoritmo

4.

```
Algoritmo punto4
    // Suponga que n es entero mayor o igual a 0
    Dimensionar j[100]
    Definir n, i Como Entero
    j[1] <- 0
    j[2] <- 1
    Leer n
    Si n >= 2 Entonces
        Para i <- 3 Hasta n+1 Hacer
            j[i] <- j[i-1] + 2*j[i-2]
        FinPara
    FinSi
    Para i <- 1 Hasta n+1 Hacer
        Escribir j[i]
    FinPara
FinAlgoritmo
```

5.

```
Algoritmo punto5
    // Suponga que n es entero positivo mayor que 1 y que la matriz A
    // solo tiene enteros
    Dimensionar A[100,100], At[100,100], res[100,100]
    Definir n, cant sup, cant inf, ac Como Entero
    Definir prom sup, prom inf Como Real
    Leer n
    Para i <- 1 Hasta n Hacer
        Para j <- 1 Hasta n Hacer
            leer A[i,j]
        FinPara
    FinPara
    prom sup <- 0
    cant sup <- 0
    Para i <- 1 Hasta n-1 Hacer
        Para j <- i+1 Hasta n Hacer
            prom sup <- prom sup + A[i,j]
            cant sup <- cant sup+1
        FinPara
    FinPara
    prom sup <- prom sup/cant sup
    prom inf <- 0
    cant inf <- 0
    Para i <- 2 Hasta n Hacer
        Para j <- 1 Hasta i-1 Hacer
            prom inf <- prom inf + A[i,j]
            cant inf <- cant inf+1
        FinPara
    FinPara
    prom inf <- prom inf/cant inf
```

```

Si prominf > promsup Entonces
    Para i<-1 Hasta n Hacer
        Para j <- 1 Hasta n Hacer
            At[i,j] = A[j,i]
        FinPara
    FinPara
    Para i<-1 Hasta n Hacer
        Escribir ""
        Para j <- 1 Hasta n Hacer
            Escribir Sin Saltar At[i,j], "|"
        FinPara
    FinPara
SiNo
    Para i<-1 Hasta n Hacer
        Para j <- 1 Hasta n Hacer
            At[i,j] = 2*A[j,i]
        FinPara
    FinPara
    Para i <- 1 Hasta n Hacer
        Para j <- 1 Hasta n Hacer
            ac <- 0
            Para k <- 1 Hasta n Hacer
                ac <- ac + At[i,k]*A[k,j]
            FinPara
            res[i,j] <- ac
        FinPara
    FinPara

    Para i<-1 Hasta n Hacer
        Escribir ""
        Para j <- 1 Hasta n Hacer
            Escribir Sin Saltar res[i,j], "|"
        FinPara
    FinPara
FinSi
FinAlgoritmo

```

6.

Algoritmo punto6

```

// Suponga que n y m son numeros enteros positivos, que dentro de
// los vectores no hay numeros negativos, todos son enteros y cada
// elemento es diferente

```

```

Dimensionar x[100], z[100], res1[100], res2[100]

```

```

Definir n, m, k, contd, contd2, l Como Entero

```

```

Leer n,m

```

```

Para i <- 1 Hasta n Hacer

```

```

    Leer x[i]

```

```

FinPara

```

```

Para i <- 1 Hasta m Hacer

```

```

    Leer z[i]

```

```

FinPara

```

```

l <- 0
Para i <- 1 Hasta n Hacer
    k <- 2
    contd <- 1
    Mientras k < x[i] Hacer
        Si x[i]%k = 0 Entonces
            contd <- contd + k
        FinSi
        k <- k+1
    FinMientras
    Para j <- 1 Hasta m Hacer
        Si contd = z[j] Entonces
            k <- 2
            contd2 <- 1
            Mientras k < z[j] Hacer
                Si z[j]%k = 0 Entonces
                    contd2 <- contd2 + k
                FinSi
                k <- k+1
            FinMientras
            Si contd2 = x[i] Entonces
                l <- l+1
                res1[l] <- x[i]
                res2[l] <- z[j]
            FinSi
        FinSi
    FinPara
FinPara
FinPara
Escribir "Amigos en X y Z respectivamente"
Para i <- 1 Hasta l Hacer
    Escribir Sin Saltar res1[i], "|"
FinPara
Escribir ""
Para i <- 1 Hasta l Hacer
    Escribir Sin Saltar res2[i], "|"
FinPara
FinAlgoritmo

```

7.

Algoritmo punto7

```

// Suponga que tanto n como m son enteros positivos y que dentro
// de X solo hay enteros no negativos
Dimensionar X[100,100], v[100]
Definir n, m, contd, k, l Como Entero
Leer n,m
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        Leer X[i,j]
    FinPara
FinPara
l <- 0

```

```

Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        k <- 2
        contd <- 1
        Mientras k < X[i,j] Hacer
            Si X[i,j]%k = 0 Entonces
                contd <- contd + k
            FinSi
            k <- k+1
        FinMientras
        Si contd = X[i,j] y X[i,j] <> 1 Entonces
            l <- l+1
            v[l] <- X[i,j]
        FinSi
    FinPara
FinPara
Para i <- 1 Hasta l Hacer
    Escribir Sin Saltar v[i], "|"
FinPara
FinAlgoritmo

```

8.

```

Algoritmo punto8
// Suponga que n es entero positivo y que dentro de a solo hay
// numeros enteros
Dimensionar V[100,100], a[100]
Definir n Como Entero
Leer n
Para i <- 1 Hasta n Hacer
    Leer a[i]
FinPara
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta n Hacer
        V[i,j] <- a[i]^(j-1)
    FinPara
FinPara
Para i <- 1 Hasta n Hacer
    Escribir ""
    Para j <- 1 Hasta n Hacer
        Escribir Sin Saltar V[i,j], "|"
    FinPara
FinPara
FinAlgoritmo

```

9.

Algoritmo punto9

```
// Suponga que n es entero positivo impar
Dimensionar M[100,100]
Definir n, contnum, i, j, i2, j2, total, c, s Como Entero
leer n
total <- n*n
c <- ((n^2)+1)/2
s <- n*c
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta n Hacer
        M[i,j] = 0
    FinPara
FinPara
i <- 1
j <- (n+1)/2
Para contnum <- 1 Hasta total Hacer
    M[i,j] <- contnum
    i2 <- i
    j2 <- j
    i <- i-1
    j <- j+1
    Si i = 0 Entonces
        i <- n
    FinSi
    Si j > n Entonces
        j <- 1
    FinSi
    Si M[i,j] <> 0 Entonces
        i <- i2 +1
        j <- j2
    FinSi
FinPara
Escribir "La suma invariante del cuadrado es: ",s
Para i <- 1 Hasta n Hacer
    Escribir ""
    Para j <- 1 Hasta n Hacer
        Escribir Sin Saltar M[i,j], "|"
    FinPara
FinPara
FinAlgoritmo
```

10.

Algoritmo punto10

```
// Suponga que n, m, p, q, son iguales pues las operaciones no se
// realizarán a menos que las matrices sean de igual dimensión y
// cuadradas
Dimensionar A[100,100], B[100,100], At[100,100], Bt[100,100],
res[100,100], res2[100,100]
Definir n, m, p, q, mayorA, mayorB, ac Como Entero
Leer n, m, p, q
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        Leer A[i,j]
    FinPara
FinPara
mayorA <- A[1,1]
Para i <- 1 Hasta p Hacer
    Para j <- 1 Hasta q Hacer
        Leer B[i,j]
    FinPara
FinPara
mayorB <- B[1,1]
Para i <- 1 Hasta n Hacer
    Para j <- 1 Hasta m Hacer
        Si A[i,j] > mayorA Entonces
            mayorA <- A[i,j]
        FinSi
    FinPara
FinPara
Para i <- 1 Hasta p Hacer
    Para j <- 1 Hasta q Hacer
        Si B[i,j] > mayorB Entonces
            mayorB <- B[i,j]
        FinSi
    FinPara
FinPara
Si mayorA > mayorB Entonces
    Escribir "el mayor se encuentra en A"
    Para i <- 1 Hasta m Hacer
        Para j <- 1 Hasta n Hacer
            At[i,j] <- A[j,i]
        FinPara
    FinPara
    Para i <- 1 Hasta n Hacer
        Para j <- 1 Hasta m Hacer
            res[i,j] <- A[i,j] + B[i,j]
        FinPara
    FinPara
    Para i <- 1 Hasta m Hacer
        Para j <- 1 Hasta m Hacer
            ac <- 0
```



```

        Para k <- 1 Hasta n Hacer
            ac <- ac + res[i,k]*At[k,j]
        FinPara
        res2[i,j] <- ac
    FinPara
FinPara
SiNo
    Escribir "el mayor se encuentra en B"
    Para i <- 1 Hasta q Hacer
        Para j <- 1 Hasta p Hacer
            Bt[i,j] <- B[j,i]
        FinPara
    FinPara
    Para i <- 1 Hasta p Hacer
        Para j <- 1 Hasta q Hacer
            res[i,j] <- B[i,j] - A[i,j]
        FinPara
    FinPara
    Para i <- 1 Hasta q Hacer
        Para j <- 1 Hasta q Hacer
            ac <- 0
            Para k <- 1 Hasta p Hacer
                ac <- ac + res[i,k]*Bt[k,j]
            FinPara
            res2[i,j] <- ac
        FinPara
    FinPara
FinSi
Para i <- 1 Hasta n Hacer
    Escribir ""
    Para j <- 1 Hasta n Hacer
        Escribir Sin Saltar res2[i,j], "|"
    FinPara
FinPara
FinAlgoritmo
11.
Algoritmo punt011
    // Suponga que n es un entero positivo y que los elementos de v
    son enteros
    Dimensionar v[100]
    Definir n, cont, x Como Entero
    Leer n
    Para i <- 1 Hasta n Hacer
        Leer v[i]
    FinPara
    Leer x
    cont <- 0
    Para i <- 1 Hasta n Hacer
        Si x = v[i] Entonces
            cont <- cont + 1
        FinSi
    FinPara
    Escribir "El elemento ",x," se repite ",cont," veces en el vector"

```

FinAlgoritmo

12.

Algoritmo punto12

```
// Suponga que n es positivo y que en el vector no hay numeros
//negativos
Dimensionar v[100], res[100]
Definir n, d, r, j, num Como Entero
Leer n
Para i <- 1 Hasta n Hacer
    Leer v[i]
FinPara
j <- 0
Para i <- 1 Hasta n Hacer
    num <- v[i]
    r <- 0
    Mientras num <> 0 Hacer
        d <- num%10
        r <- r*10 + d
        num <- trunc(num/10)
    FinMientras
    Si v[i] = r Entonces
        j <- j+1
        res[j] <- v[i]
    FinSi
FinPara
Escribir "Hay ",j, " numeros palindromos"
Para i <-1 Hasta j Hacer
    Escribir Sin Saltar res[i], "|"
FinPara
```

FinAlgoritmo

13.

Algoritmo punto13

```
// Suponga que n es entero positivo
Dimensionar v[100]
Definir n, x, n2 Como Entero
Leer n
n2 <- n
Para i <- 1 Hasta n Hacer
    Leer v[i]
FinPara
leer x
Para i <-1 Hasta n Hacer
    Mientras x = v[i]
        Para j <- i Hasta n-1 Hacer
            v[j] <- v[j+1]
        FinPara
        n2 <- n2-1
    FinMientras
FinPara
Para i <-1 Hasta n2 Hacer
    Escribir Sin Saltar v[i], "|"
FinPara
```

FinAlgoritmo