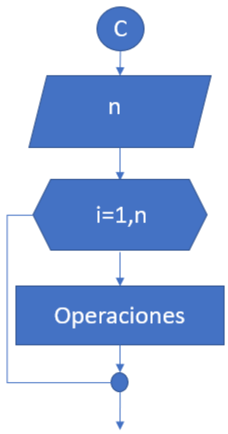
BUCLES SECUENCIALES: VECTORES Y MATRICES

Bucles Secuenciales

Se trata de una estructura de algoritmia que nos permite aplicar la repetición de un cierto conjunto de operaciones un número predeterminado de veces.

Para ello empleamos una variable de control del bucle, que se suele denotar con una letra (i, j, k, etc.), y que va recorriendo un conjunto de previamente declarado de valores en un orden concreto. Para cada valor que toma la variable de control en cada vuelta que da el bucle, se ejecuta una vez el mismo conjunto de operaciones.

El bucle secuencial se representa en el organigrama con una forma hexagonal. La estructura general de un bucle secuencial es:



<u>Pseudo-código</u>

Inicio

  Leer n

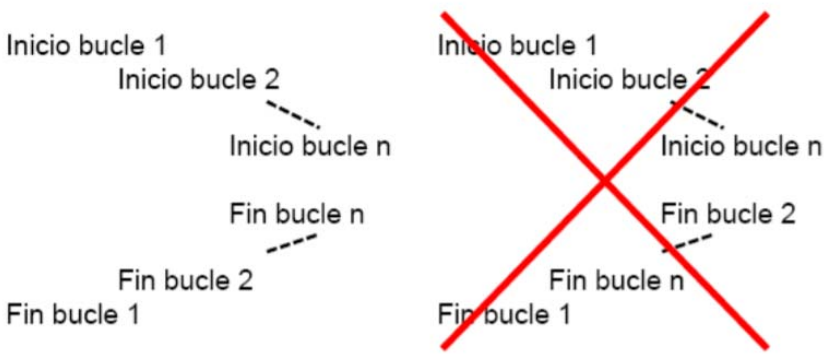
  Para i desde 1 hasta n hacer,

    Operaciones

  Fin del bucle en i

Fin

Los bucles secuenciales pueden superponerse unos dentro de otros en un mismo organigrama creando lo que se conoce como bucles anidados o *nested loops.* Pero se debe cumplir siempre que:



Los bucles secuenciales son estructuras que dentro del algoritmia se emplean muy comúnmente para resolver sumatorios y productorios (mirar apuntes de *Sumatorio y Productorio*) y también para trabajar con matrices y vectores, de los que hablaremos en este mismo recurso.

Vectores y matrices

Vectores y matrices son variables subindicadas que se emplean en algoritmia para almacenar más de un valor en una variable. Por un lado, los vectores son variables que almacenan valores en una estructura lineal. Mientras tanto, las matrices son variables que almacenan valores componiendo una estructura con filas y columnas.

La mejor forma de comprender como se emplean estas variables en programación es realizando algunos ejemplos que nos permitan conseguir una idea general de como se trabajan este tipo de variables en algoritmia.

Ejemplo 1: Realizar un organigrama que calcule el producto escalar de dos vectores X[i] y Y[i]

Teniendo dos vectores X=(x_1, x_2,...x_n), Y=(y_1, y_2,...y_n)

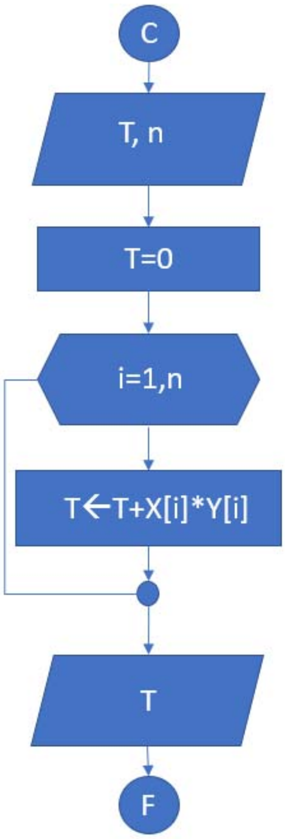
su producto escalar sería: X*Y= x_1*y_1+x_2*y_2+...+x_n*y_n

En un organigrama podemos almacenar el valor del producto escalar en una variable que llamaremos T. Además, podemos representar esta misma operación como un sumatorio:

$$T = \sum_{i=1}^{n} x[i] * y[i]$$

De esta forma, podremos usar la estructura del bucle secuencial para resolver esta operación como haríamos con un sumatorio. Es IMPORTANTE tener en cuenta entonces, que para resolver un sumatorio debemos inicializar la variable T a cero.

Para crear el organigrama solo tendremos que introducir los datos que debemos aportar al algoritmo para iniciar el proceso. Después inicializamos esa variable T, en la que almacenaremos el valor del producto escalar, a cero. Entramos en el bucle secuencial, en el que la variable i irá tomando valores desde 1 hasta n. Para cada valor que tome la variable i se realizará la operación P←P+X[i]*Y[i]. Esta operación expresa que, para cada vuelta que da el bucle, se sumará al valor almacenado el T el producto de los valores que toman X e Y en la posición i correspondiente. Tras realizar la operación respectiva al valor n, el algoritmo sale del bucle. Finalmente, una vez fuera del bucle, el algoritmo aporta el valor de T.



Ejemplo 2: Realizar un organigrama que multiplique a dos matrices A[m, p] y B[p, n] para obtener una matriz C[m, n].

Como sabemos, gracias a lo aprendido sobre matrices, para poder calcular el producto entre dos matrices, el número de columnas de la matriz A debe ser igual al número de filas de la matriz B, (p). Como resultado, la matriz C es una matriz con el número de filas de la matriz A y el número de columnas de la matriz B.

De esta forma, para poder calcular esta operación con un algoritmo, los valores que tome la variable k deben ir a la par tanto en las columnas de la matriz A como en las filas de la matriz B. Mientras tanto, la variable i toma valores para las filas de la matriz A y la variable j toma valores para las columnas de la matriz B.

Por lo tanto, podremos expresar este producto entre matrices de la siguiente forma:

$$C[i,j] = \sum_{k=1}^{p} A[i,k] * B[k,j]$$

(i=1,…m; j=1,…n)

Podemos observar que el producto entre matrices se puede representar como un sumatorio. Además, para desarrollar este algoritmo necesitaremos de una estructura de bucles anidados.

*Recordamos: hay que inicializar la función C a cero antes de entrar en el bucle en el que vayamos a realizar el cálculo.