

Arreglos

Recuerdas que solíamos almacenar un valor en una variable:

$$a \leftarrow 15$$

y de pronto deseamos almacenar varios valores en una sola variable:

$$a \leftarrow 15, 19, 14, 7, 8$$

$$1 \quad 2 \quad 3 \quad 4 \quad 5$$

En este caso el 15 es el primer valor, el 19 el segundo valor, el 14 el tercer valor y así sucesivamente. Y de pronto te sale, que se usa una variable que almacena un conjunto de valores.

| | | | | |
|----|----|----|---|---|
| a | | | | |
| 15 | 19 | 14 | 7 | 8 |
| 1 | 2 | 3 | 4 | 5 |

Arreglos - Definición

Un arreglo es una colección finita, homogénea y ordenada de elementos, es finita debido a que es una estructura estática, por lo que su dimensión debe ser declarada con anterioridad.

Los arreglos pueden ser unidimensionales o de dimensión 1, es decir, manejan un índice para hacer referencia a cada elemento del arreglo.

Los arreglos pueden ser bidimensionales o de dimensión 2, es decir, manejan dos índices para hacer referencia a cada elemento del arreglo.

Los arreglos, pueden ser tridimensionales o de dimensión 3, es decir, manejan tres índices para hacer referencia a cada elemento del arreglo.

Por último, los arreglos pueden ser de dimensión n, por lo que, manejan n índices para hacer referencia a cada elemento del arreglo.

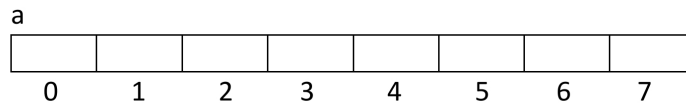
A continuación, se muestra la estructura de datos:



Vector

Es un arreglo unidimensional, finito, estático, que almacena valores del mismo tipo de datos. Se debe dimensionar antes de usar un vector, dando un tamaño aproximado según la cantidad de datos a manejar. La dimensión de cada vector involucra un espacio de memoria, por lo que se recomienda no usar indiscriminadamente la definición de vectores o arreglos.

Ejemplo: `dim a (7)` //Se dimensiona un vector de tamaño 7



El vector se define desde la posición 0, 1, 2, 3, ..., sin embargo, en adelante usaremos desde la posición 1 para no confundir la cantidad de elementos y esta sea exacta, ya que el 0 aumenta un elemento adicional al conteo natural.

Entrada

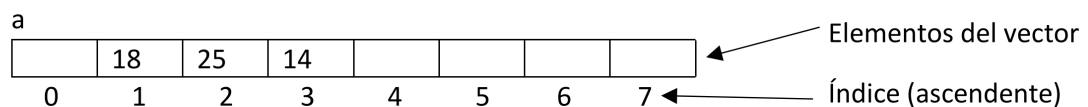
Una vez definido el vector, se procede a llenar con elementos, uno por uno, es decir, podríamos indicar lo siguiente: `dim a[7]`

Leer `a[1]` // entrada de datos para el primer elemento (18)

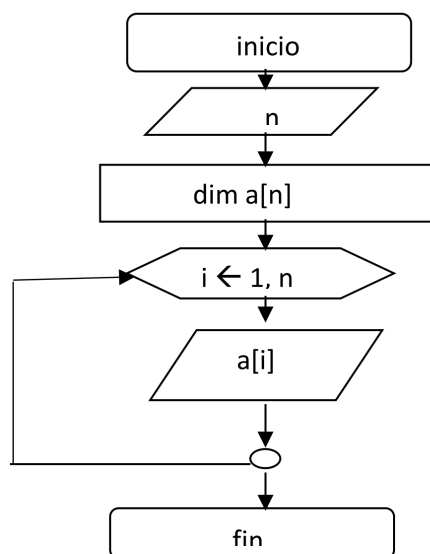
Leer `a[2]` // entrada de datos para el segundo elemento (25)

Leer `a[3]` // entrada de datos para el tercer elemento (14) y así sucesivamente.

Por lo tanto, es posible usar un ciclo para llenar de valores el vector de tamaño `n`.



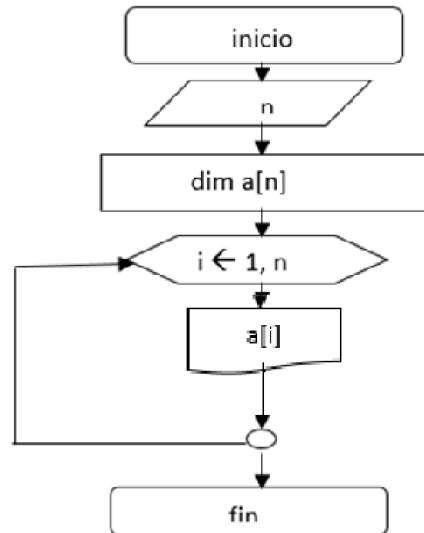
El diagrama de flujo es el siguiente



Salida.

Para mostrar un vector, se debe mostrar elemento por elemento, por lo tanto, es muy parecido al anterior proceso, un ciclo y mostrar uno por uno los elementos del vector. Veamos un ejercicio.

El diagrama de flujo es el siguiente



Ejercicio 1. Dado un vector $a(n)$, llenar con elementos enteros, y rotar los elementos pares a la derecha.

Por ejemplo:

Entrada:

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| a | 17 | 16 | 21 | 12 | 10 | 15 | 24 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Salida:

| | | | | | | | |
|---|----|----|----|----|----|----|----|
| a | 17 | 24 | 21 | 16 | 12 | 15 | 10 |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

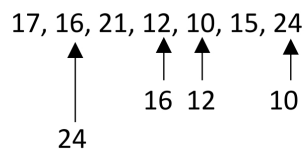
Primer paso dimensionar el vector

Segundo paso llenar el vector (entrada de datos)

Rotar solo los elementos pares

Mostrar el vector

Rotar los elementos en el mismo vector, veamos:



La salida: 17, 24, 21, 16, 12, 15, 10

Recorremos buscando el primer elemento par (Si $a[i] \% 2 = 0$), si es par, en el ejemplo el 16, guardamos ambos (elemento 16 y el índice 2). Buscamos el siguiente par, en el ejemplo el 12, cambiamos (guardo elemento 12 y reemplazo por anterior 16), y así sucesivamente. El último elemento par, el 24 lo almaceno en el primer índice que se guardó, en 2.