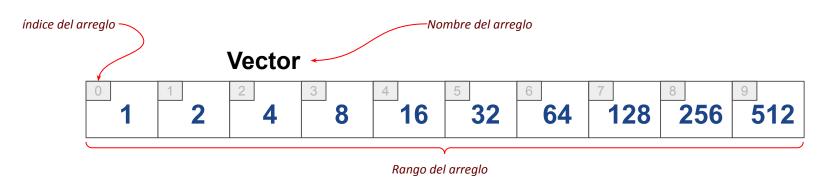


Arrays Operations

[Assign | Print | Update | Order | Search]

Conociendo un Array





<u>Declaración de un Array</u>

Integer Vector[10];

- Estamos declarando un arreglo que contiene <u>10 elementos de tipo entero</u>, las cuales pueden ser accedidas mediante el nombre **Vector**.
- ☐ Cada variable tiene un índice (dirección en memoria), que siempre se inicia en 0 y termina en 'i -1', donde 'i' representa la cantidad de elementos que tiene el arreglo.

AED

Imprimir un dato de un Array



Vector



<u>Imprimir valores de los elementos de Array</u>

Print Vector[3];

8

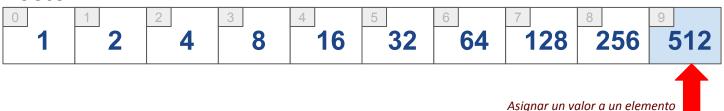
- ☐ Print nos permite obtener el valor de la variable almacenada en el <u>índice 3</u> del arreglo Vector.
- ☐ El resultado nos imprime 8.

AED

Asignar valores a los elementos de un Array







Asignar valores a un Array

AED

 $Vector[9] \leftarrow$ **512**;

- Esta declaración nos permite asignar el valor 512 al elemento cuyo <u>índice es 9</u>.
- ☐ El resultado es que se carga el valor en el espacio de memoria que se indica a continuación.

Crear un pseudocódigo para este arreglo



Vector



```
Pseudocódigo DataArray
integer Vector[10];
Var i integer;
i = 0:
Start
While (i < 10) do
      Vector[i] \leftarrow 2^{**i}
      Print Vector[i]
      i = i + 1
End while
End
```

Crear un pseudocódigo la serie de Fibonacci



Vector



```
Pseudocódigo SerieFibonacci
                                                         Pseudocódigo SerieFibonacci
integer fibonacci[10];
                                                         integer fibonacci[10];
Var i integer;
                                                          Var i integer;
i = 2;
                                                         i = 0:
Fibonacci[0] \leftarrow 0 Fibonacci[1] \leftarrow 1
                                                         Fibonacci[0] ← 0 Fibonacci[1] ← 1
Start
                                                          Start
Do
                                                          While (i < 10) Do
      fibonacci[i] ← fibonacci[i-1] + fibonacci[i-2]
                                                               fibonacci[i+2] ← fibonacci[i+1] + fibonacci[i]
      Print fibonacci[i]
                                                                Print fibonacci[i]
                                                               i = i + 1
      i = i + 1
While (i < 10)
                                                         End while
End
                                                          End
```