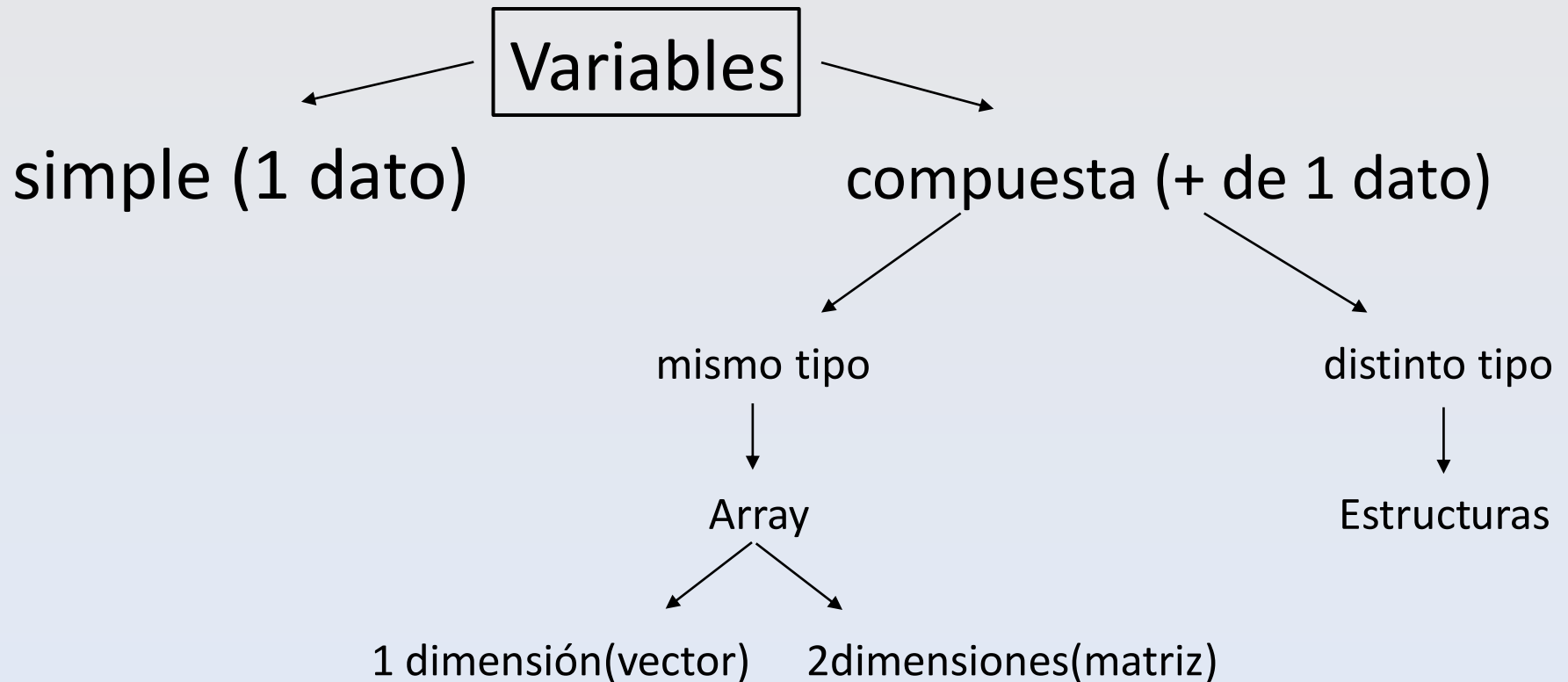




CLASE 5

Arrays:



CLASE 5

Arrays: Definición

- Un vector o array es un conjunto de elementos del mismo tipo (enteros, flotantes, caracteres)
- Se agrupan bajo un mismo nombre
- Se diferencian entre sí por un índice.

CLASE 5

Arrays: Declaración

La forma genérica de declararlo en un programa es la siguiente:

Tipo nombre[cantidad]

Tipo: tipo de dato(*int, float, char*, etc.)

nombre: nombre del array

Cantidad: cantidad de elementos que contiene el array.

CLASE 5

Arrays:

Ejemplo:

```
int v[5];
```

Declaración de un array que se llama v de 5 enteros

```
float vec[10];
```

Declaración de un array que se llama vec de 10 flotantes

```
char v1[5];
```

Declaración de un array llamado v1 de 5 caracteres

CLASE 5

Arrays:

int v[5];

Representación gráfica del array “v” puede ser:

Valor	12	23	9	37	31
Índice	0	1	2	3	4

Como se puede ver el índice comienza siempre en cero y va hasta la cantidad de elementos menos uno.

CLASE 5

Arrays: Elementos

```
int v[5];
```

Para acceder a un elemento cualquiera de array, por ejemplo al tercer elemento se escribe:

```
v[2]
```

Donde “v” es el nombre del array y entre corchetes se le pone el índice (posición del elemento).

CLASE 5

Arrays:

int v[5];

Valor	12	23	9	37	31
Índice	0	1	2	3	4

El array tiene 5 enteros, el nombre es “v” y sus elementos son:

- v[0] tiene el valor 12 y es el primer elemento del array
- v[1] tiene el valor 23 y es el segundo elemento del array
- v[2] tiene el valor 9 y es el tercer elemento del array
- v[3] tiene el valor 37 y es el cuarto elemento del array
- v[4] tiene el valor 31 y es el quinto elemento del array

-Entre el número de elemento y el índice existe una diferencia de 1

-v[5]no existe como elemento del array.

CLASE 5

Arrays: Carga secuencial

En la carga secuencial se carga el array ordenadamente, es decir el primero después el segundo, el tercero y así hasta llegar al último.

La carga secuencial es ordenada y de acuerdo a esto deberemos llenar el array respetando el orden.

$v[0], v[1], v[2], v[3], v[4]$

CLASE 5

Arrays: Carga secuencial

Ejemplo que nos permite cargar un array en forma secuencial:

```
#include <stdio.h>

void main(void)
{int v[5],i;
    for (i=0;i<5;i++)
        {printf("Ingrese valor a cargar en el array ");
          scanf("%d",&v[i]);}
}
```

CLASE 5

Arrays: Carga secuencial

Ejercicio:

De los 10 empleados de una fábrica se registra:

índice	0	1	2	3	4	5	6	7	8	9
legajo	0	1	2	3	4	5	6	7	8	9
edad										
sueldo										

CLASE 5

Arrays: Carga secuencial

Ejercicio:

De los 10 empleados de una fábrica se registra:

- número de legajo (coincidente con el índice)
- edad
- sueldo

Se pide ingresar los datos consecutivamente y calcular el sueldo promedio. (Clase5-Ej1.c)

CLASE 5

Arrays: Carga aleatoria

En este caso se necesitan ingresar 2 valores, la posición donde se quiere ingresar el dato y el dato.

Puede que no se quieran cargar todos los elementos del Array, por eso conviene INICIALIZARLO. (Clase5_Ej2)

CLASE 5

Arrays: Carga aleatoria

Ejercicio:

De los 100 empleados se registra número de legajo (coincidente con el índice), edad y salario. (Clase5-Ej3.c)

- Ingresar los datos

- Calcular el sueldo promedio.

El análisis del problema se plantea exactamente igual al caso de carga secuencial. La única diferencia es que ahora se debe ingresar el legajo quien va a ser el que me indique la posición en donde se va a cargar el sueldo y la edad.

Recordar: la relación entre el índice y el legajo es $\text{legajo} = \text{índice} + 1$.

CLASE 5

Arrays: Menor y mayor

Ejercicio:

Pedir al usuario que ingrese 5 números y los guardamos en las posiciones del array.

Clase5-Ej4.c

CLASE 5

Arrays: Búsqueda

Para buscar un número dentro de un array, lo debemos recorrer y preguntar si el elemento del array es igual al número ingresado.

Clase5-Ej5.c

CLASE 5

Arrays: Búsqueda

Método de burbujeo

Una de las formas más usadas para ordenar un array se llama Método de burbujeo.

Se van comparando los elementos del array ($\text{vec}[i]$) con el resto de los elementos del array (desde la posición siguiente a “i” hasta el final del mismo).

CLASE 5

Arrays: Búsqueda

Método de burbujeo

Se utiliza un primer bucle “for” que permite iterar el array y obtener sus elementos, desde el comienzo hasta el final (utilizando “i”)

para cada elemento obtenido:

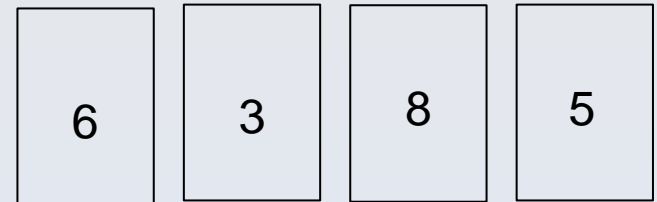
se itera nuevamente el array desde la posición siguiente, hasta el final, con un segundo bucle “for” con una variable “j” que comienza valiendo $i+1$.

CLASE 5

Arrays: Búsqueda

Método de burbujeo–ORDENAR DE MANERA ASCENDENTE

```
for(i=0;i<FIL-1;i++)  
    { for(j=i+1;j<FIL;j++)  
        {if(vec[i]>vec[j])  
            {aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux;}}}
```



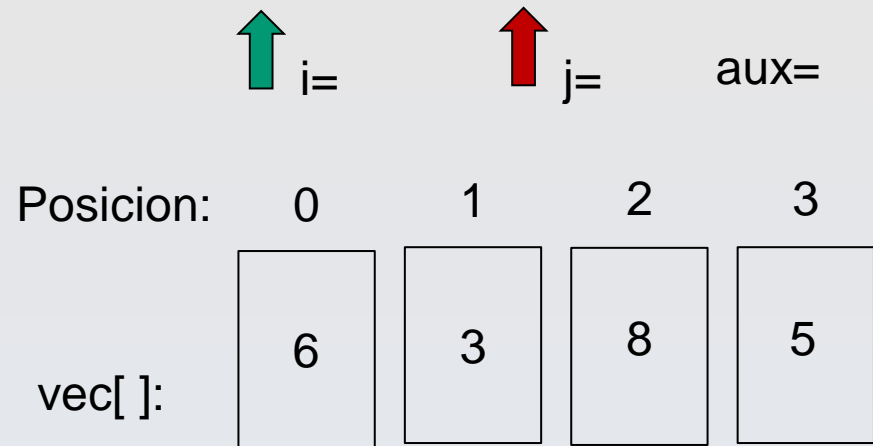
Swap

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

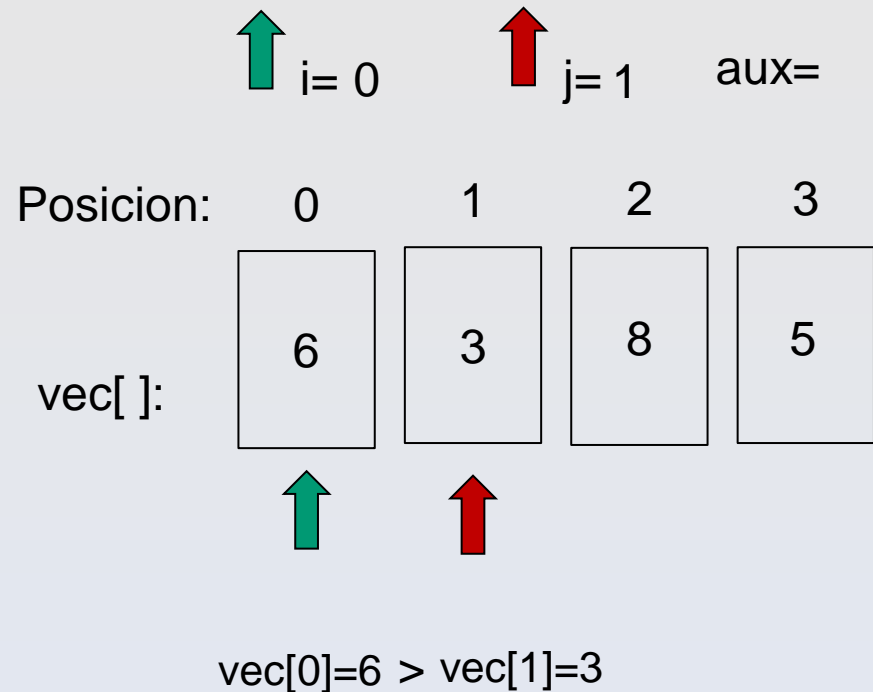


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        → { if(vec[i]>vec[j])  
                { aux=vec[i];  
                  vec[i]=vec[j];  
                  vec[j]=aux; } } }
```

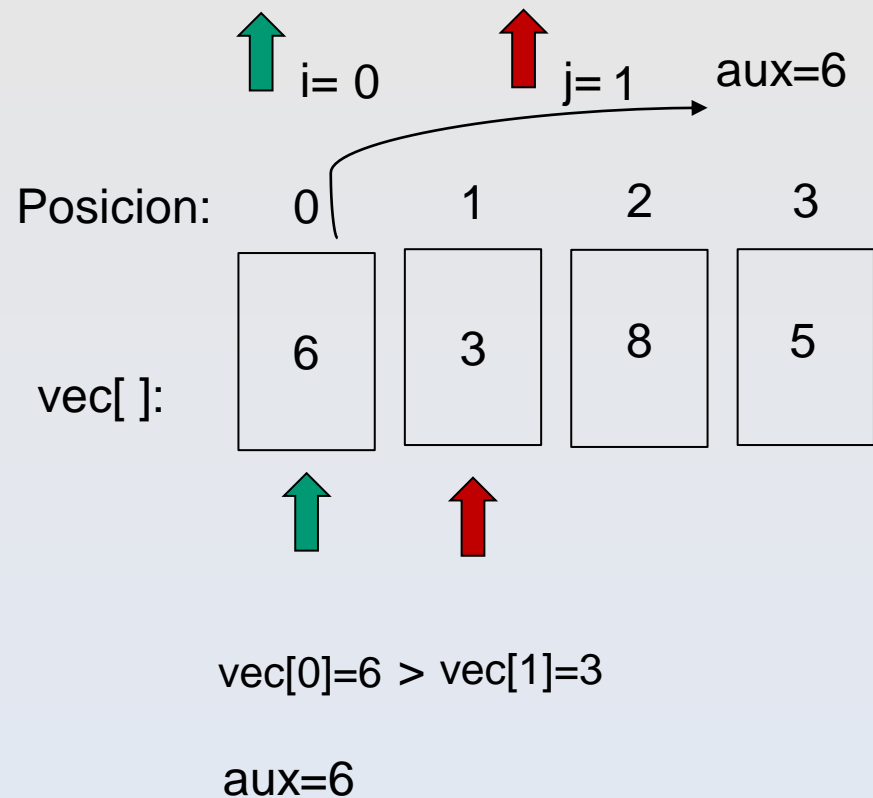


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            → { aux=vec[i];  
                vec[i]=vec[j];  
                vec[j]=aux; } } }
```

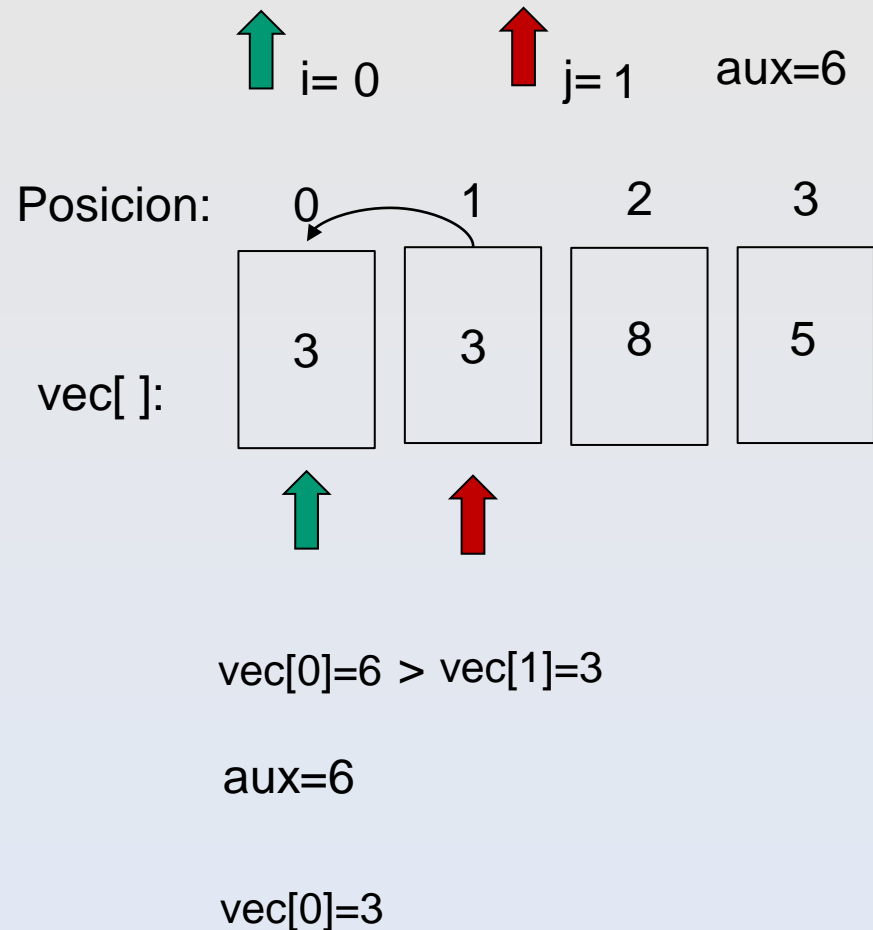


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              → vec[i]=vec[j];  
                vec[j]=aux; } } }
```

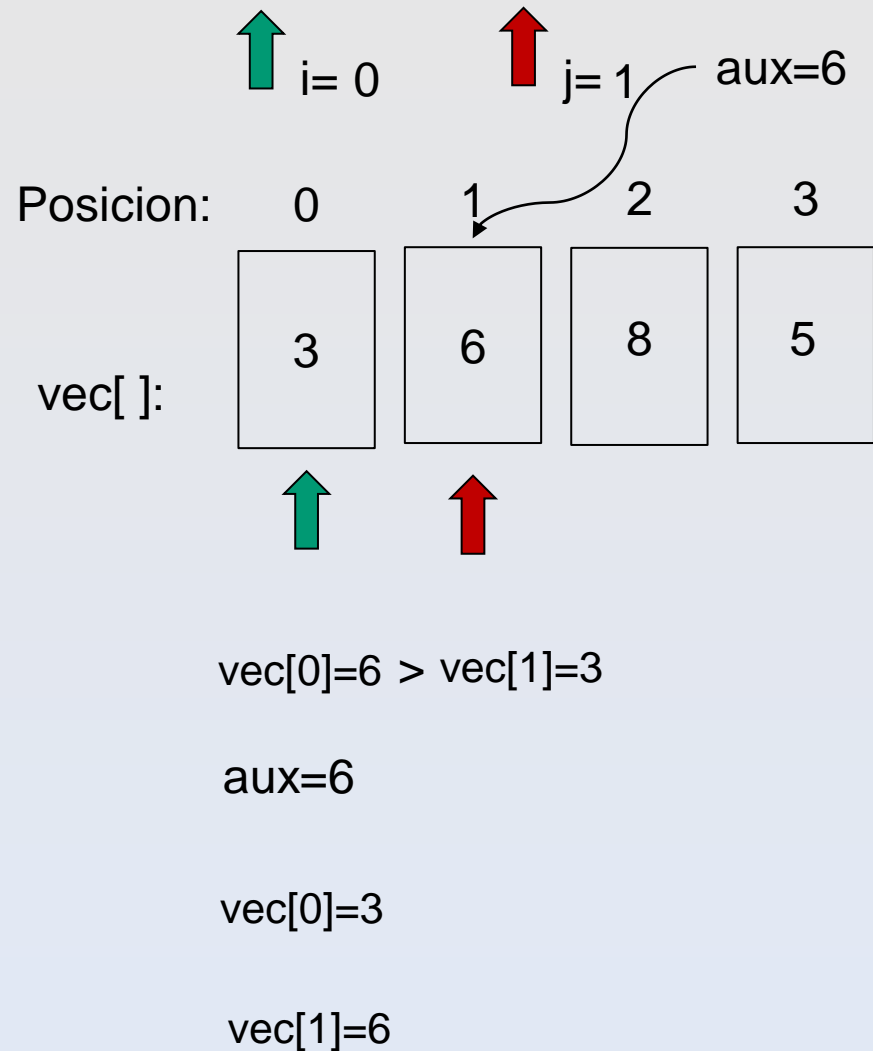


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              → vec[j]=aux; } } }
```

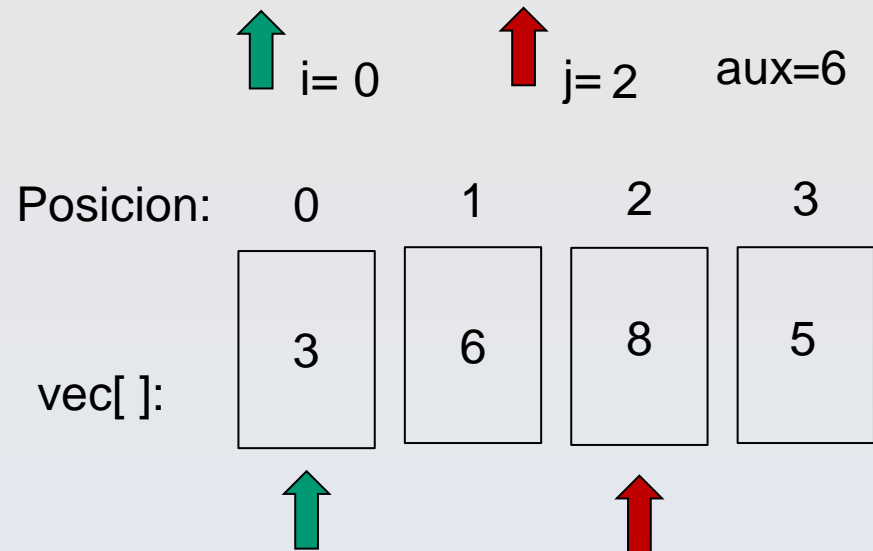


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

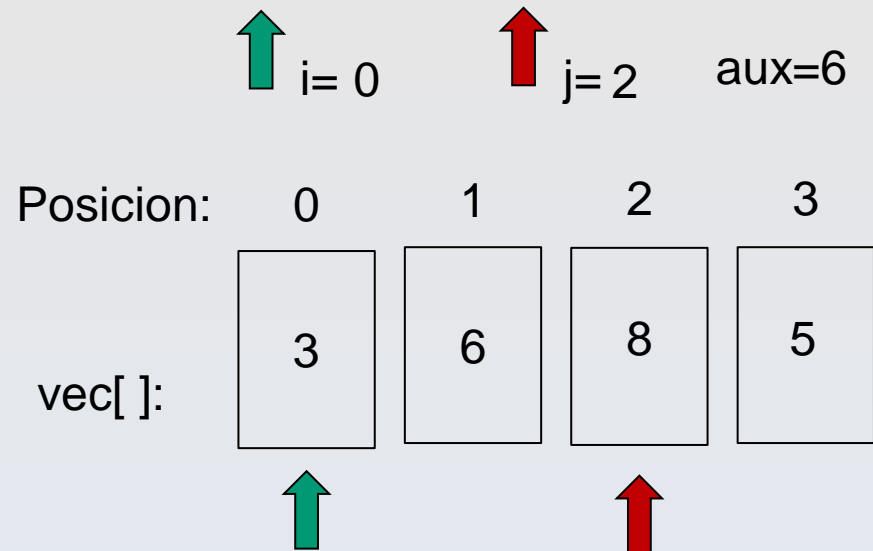


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

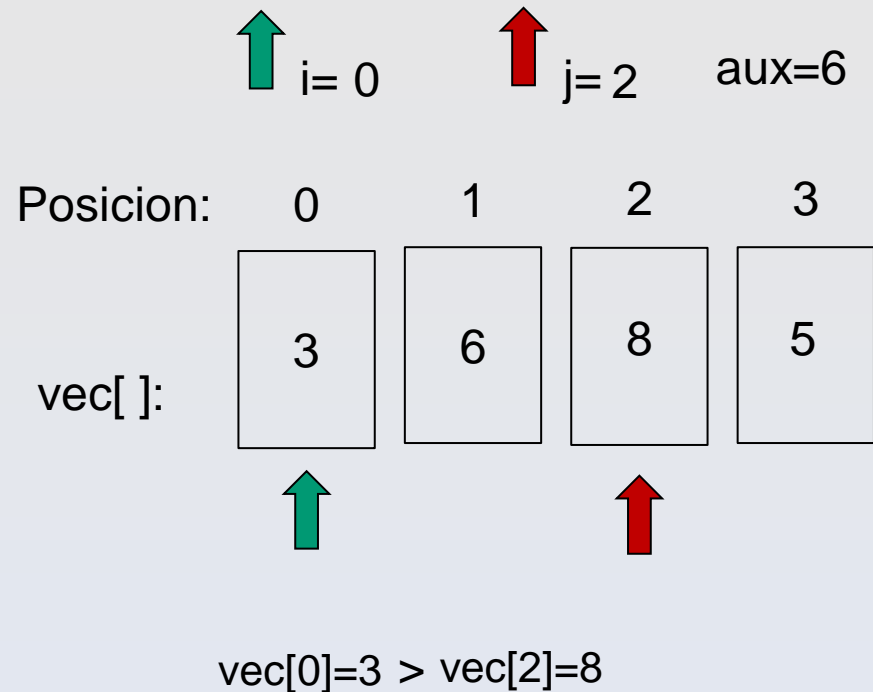


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        → { if(vec[i]>vec[j])  
                { aux=vec[i];  
                  vec[i]=vec[j];  
                  vec[j]=aux; } } }
```

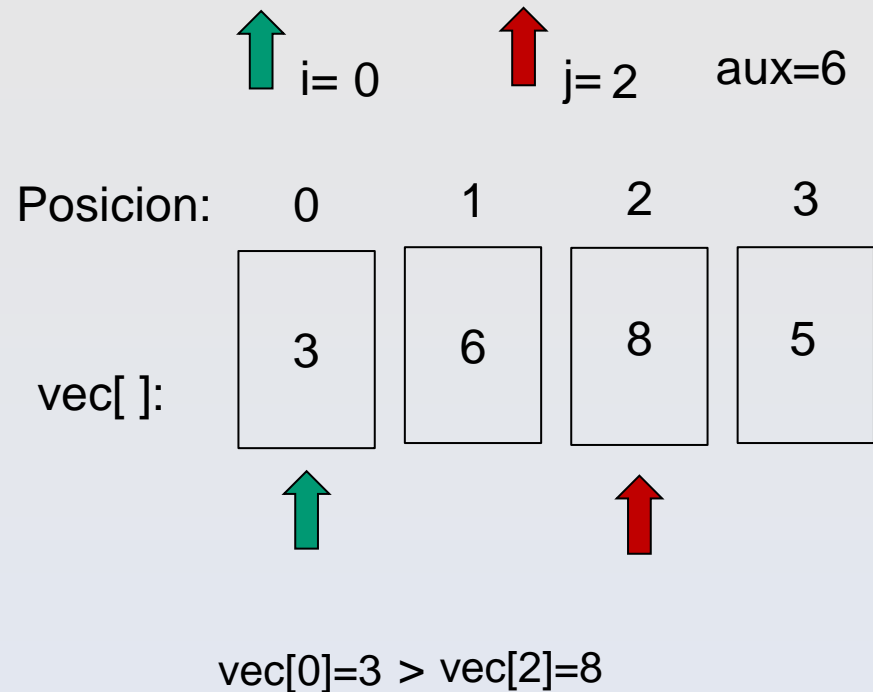


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

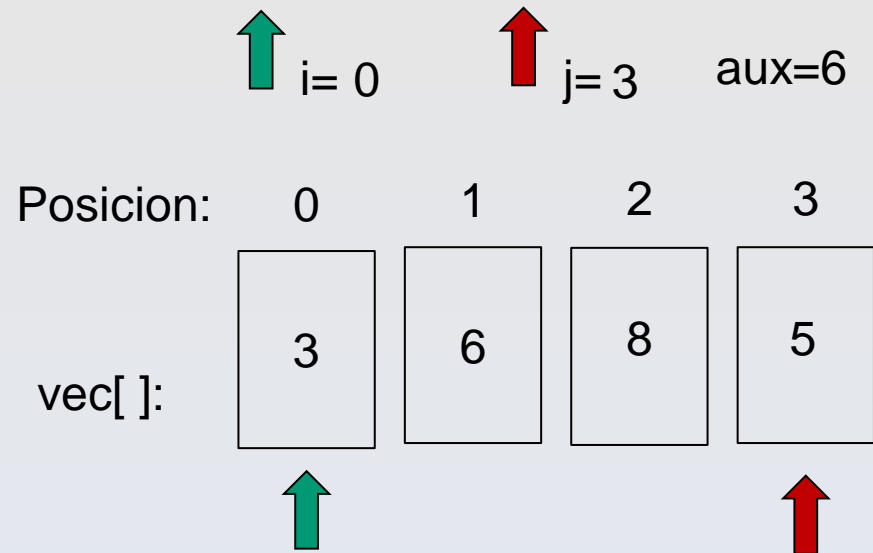


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

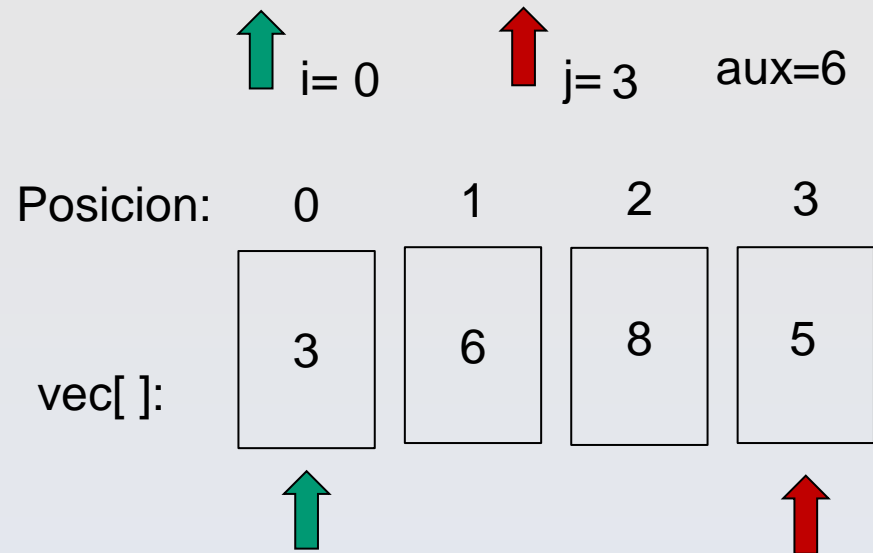


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

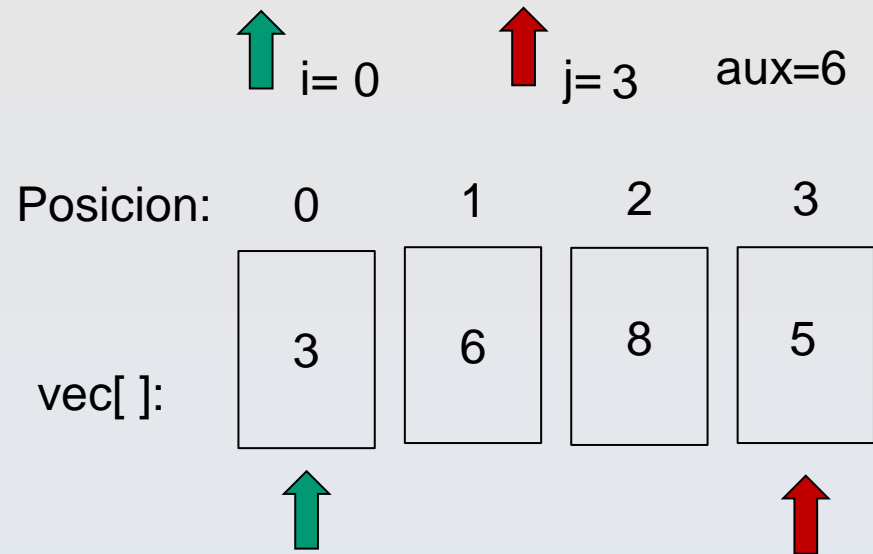


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

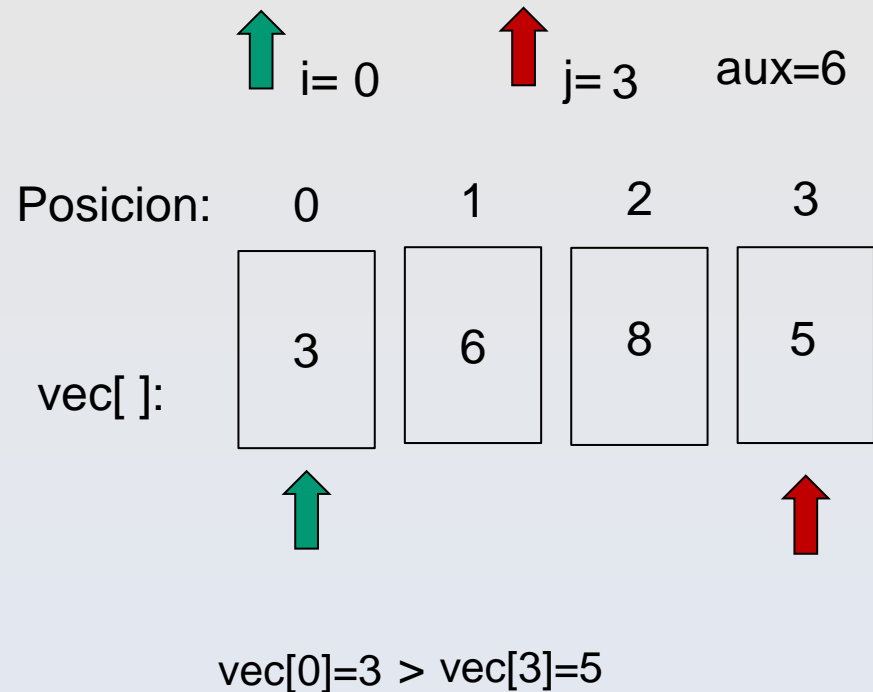


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        → { if(vec[i]>vec[j])  
                { aux=vec[i];  
                  vec[i]=vec[j];  
                  vec[j]=aux; } } }
```

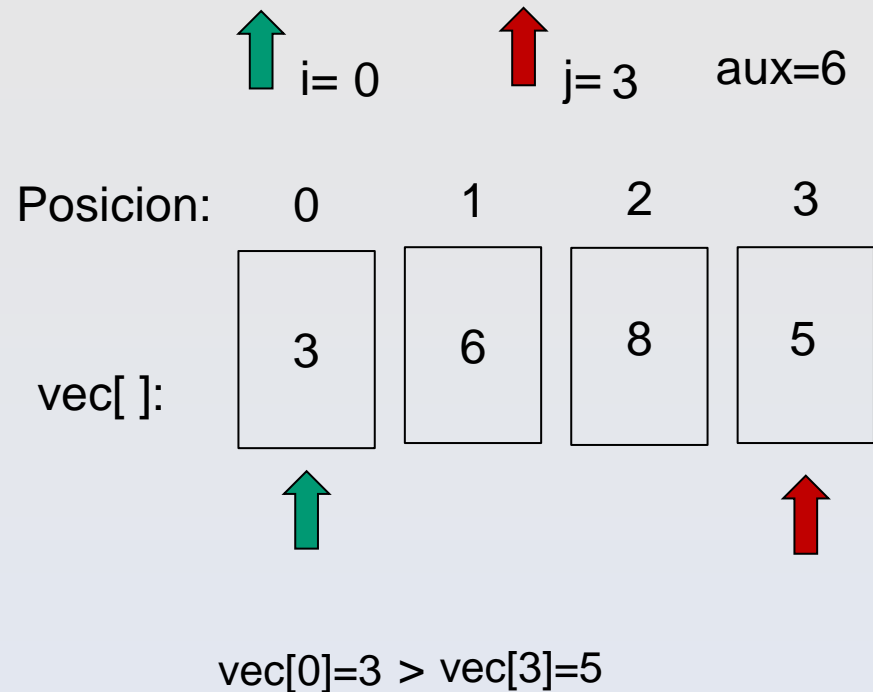


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

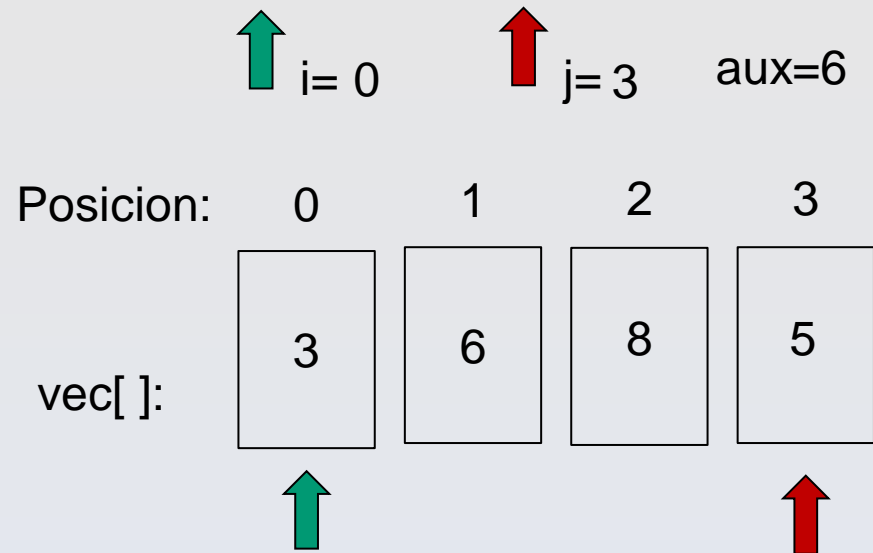


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

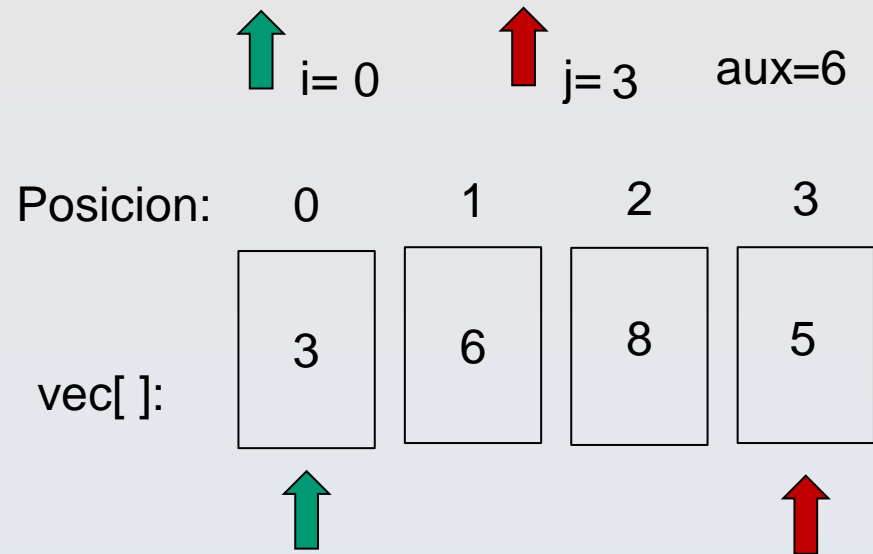


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
for(i=0;i<FIL-1;i++)  
  { for(j=i+1;j<FIL;j++)  
    { if(vec[i]>vec[j])  
      { aux=vec[i];  
        vec[i]=vec[j];  
        vec[j]=aux;}}}
```

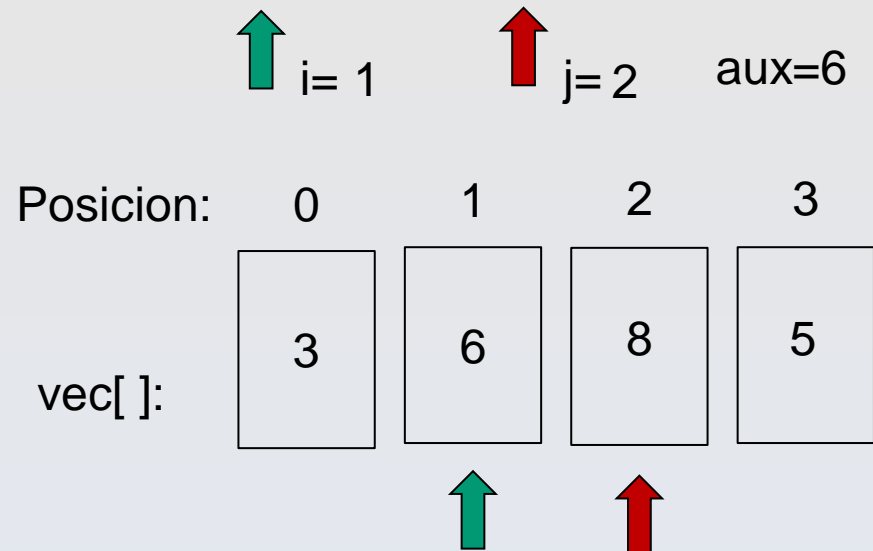


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

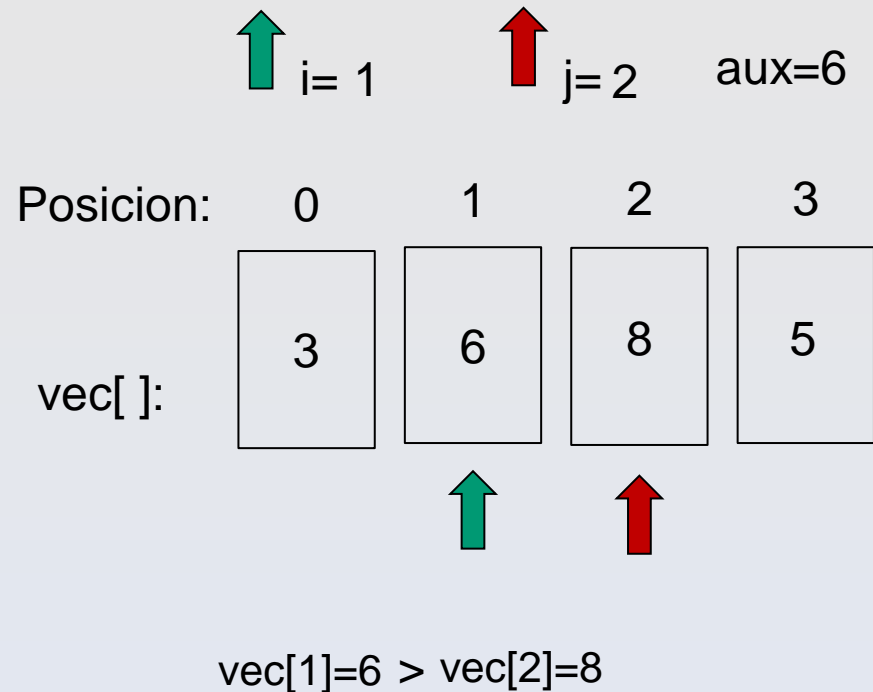


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

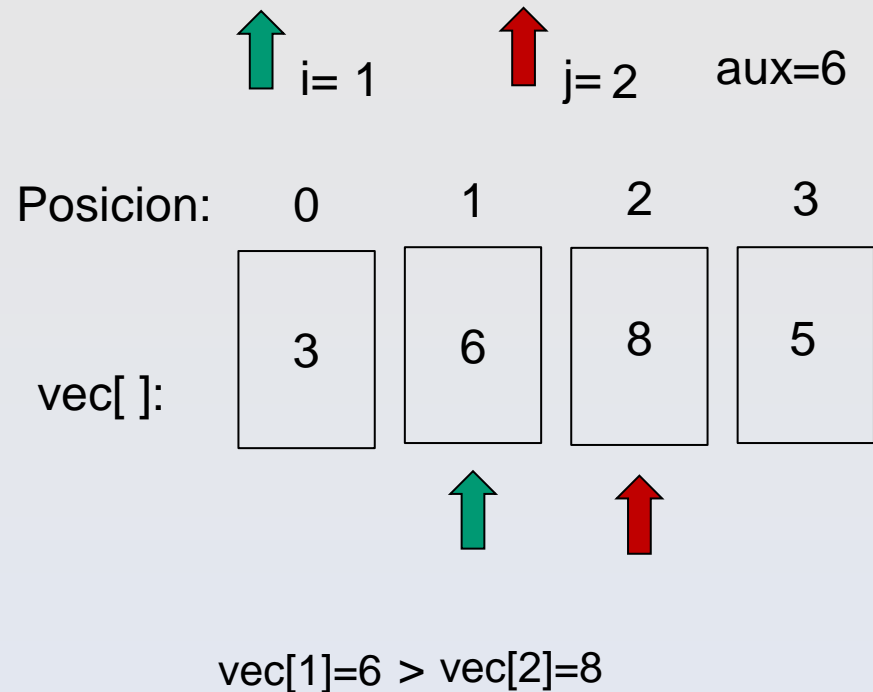


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

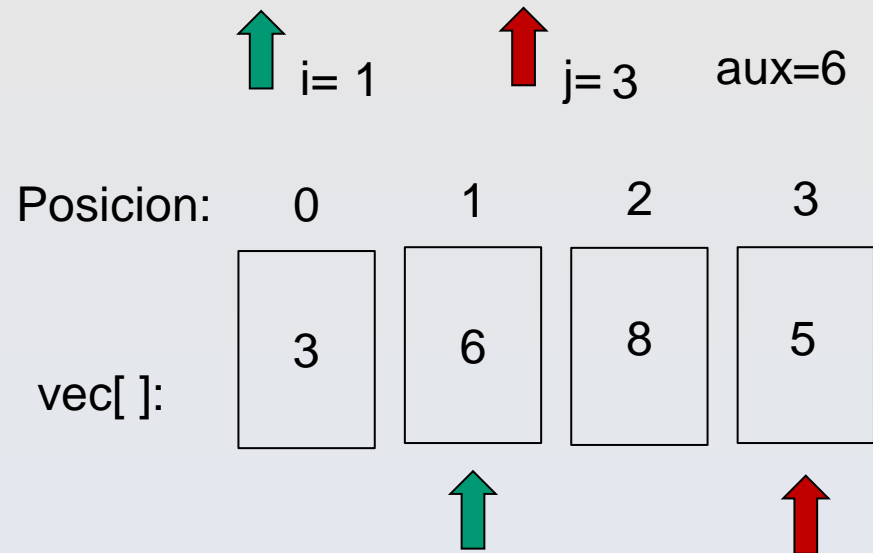


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

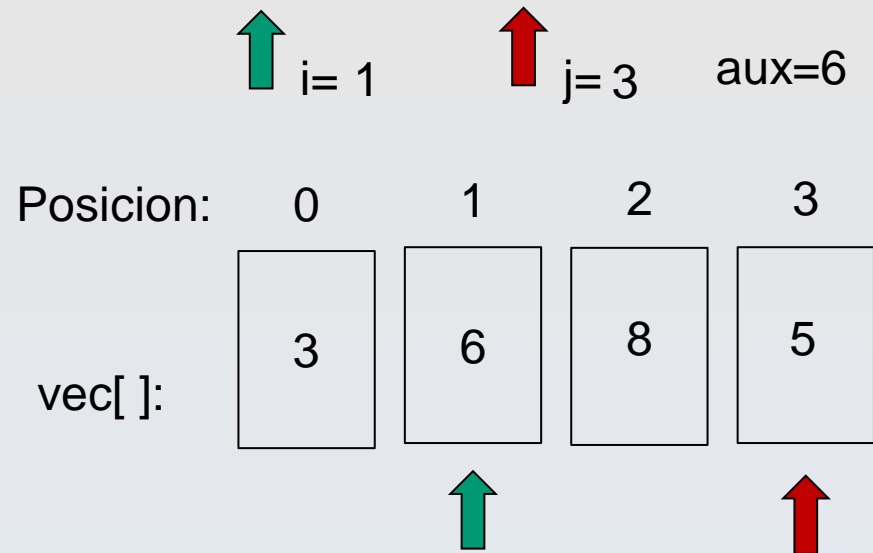


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

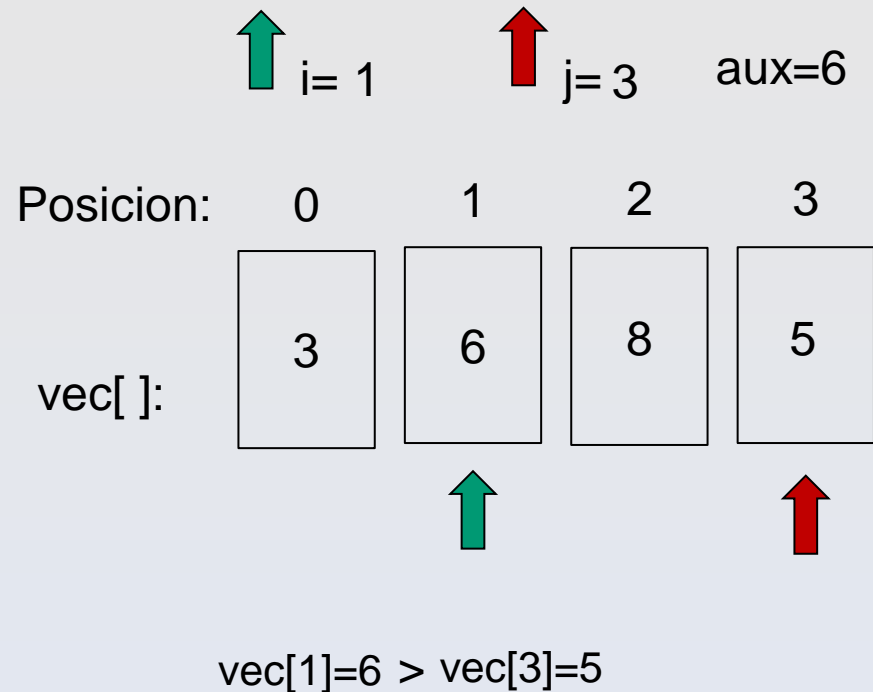


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        → { if(vec[i]>vec[j])  
                { aux=vec[i];  
                  vec[i]=vec[j];  
                  vec[j]=aux; } } }
```

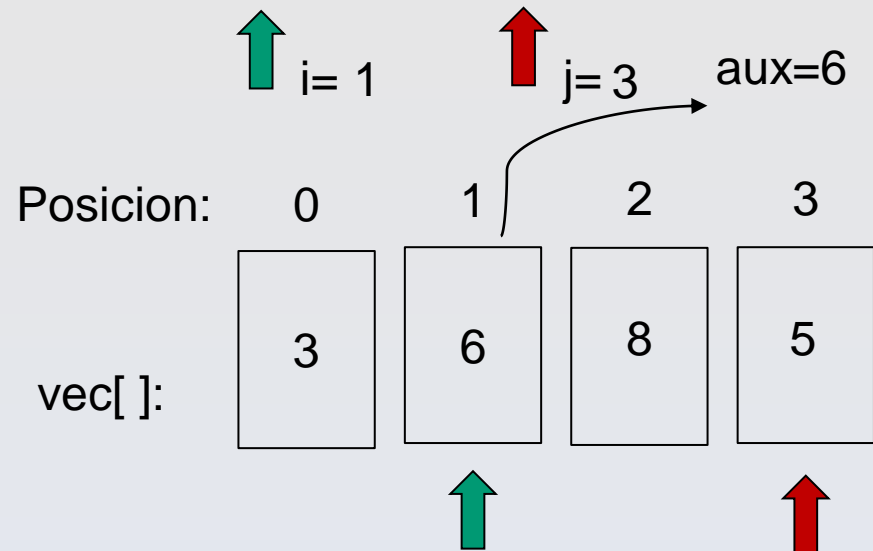


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            → { aux=vec[i];  
                vec[i]=vec[j];  
                vec[j]=aux; } } }
```



$vec[1]=6 > vec[3]=5$

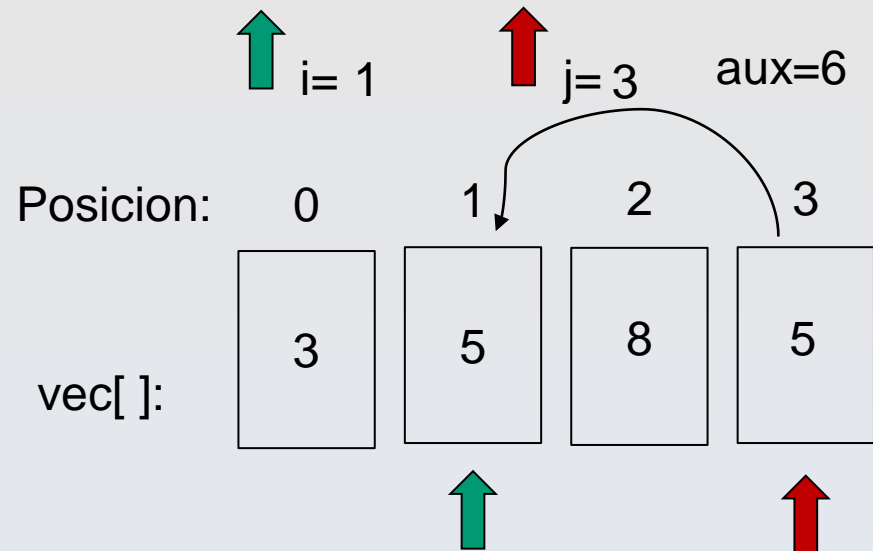
$aux=6$

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              → vec[i]=vec[j];  
                vec[j]=aux; } } }
```



$vec[1]=6 > vec[3]=5$

$aux=6$

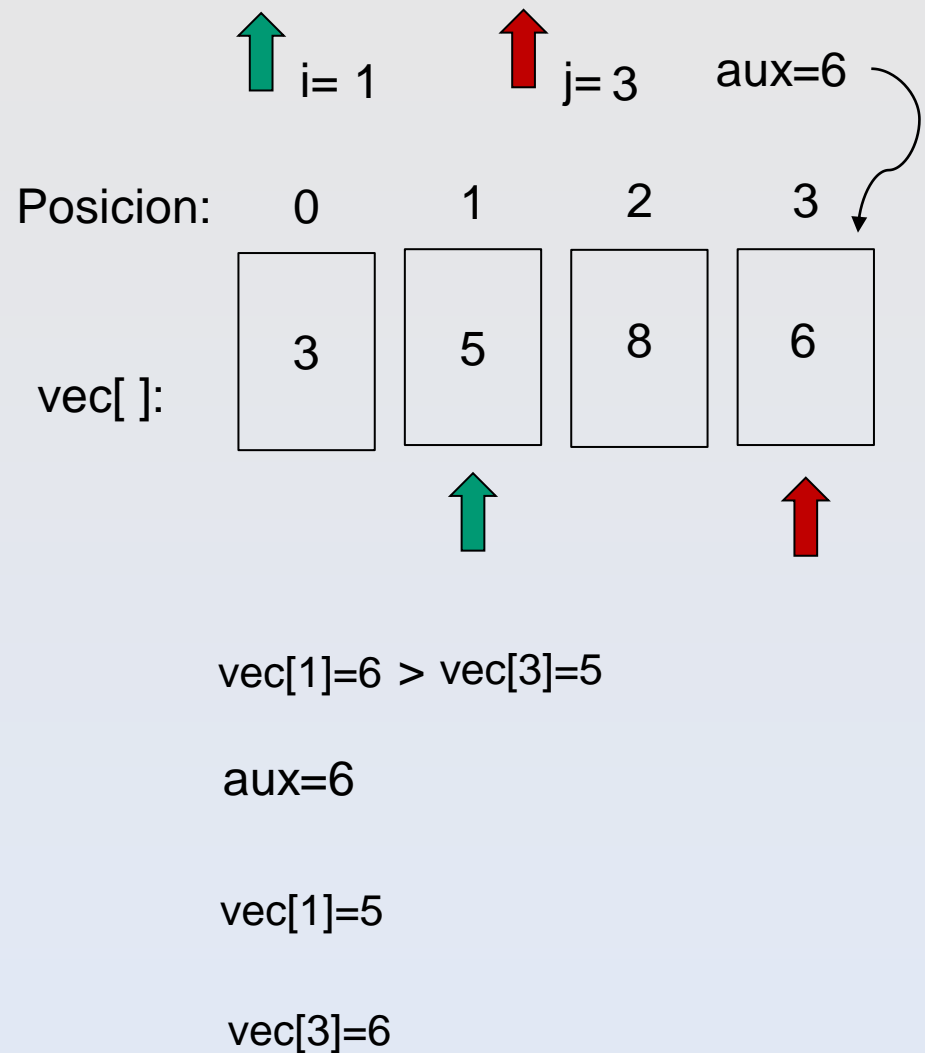
$vec[1]=5$

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              → vec[j]=aux; } } }
```

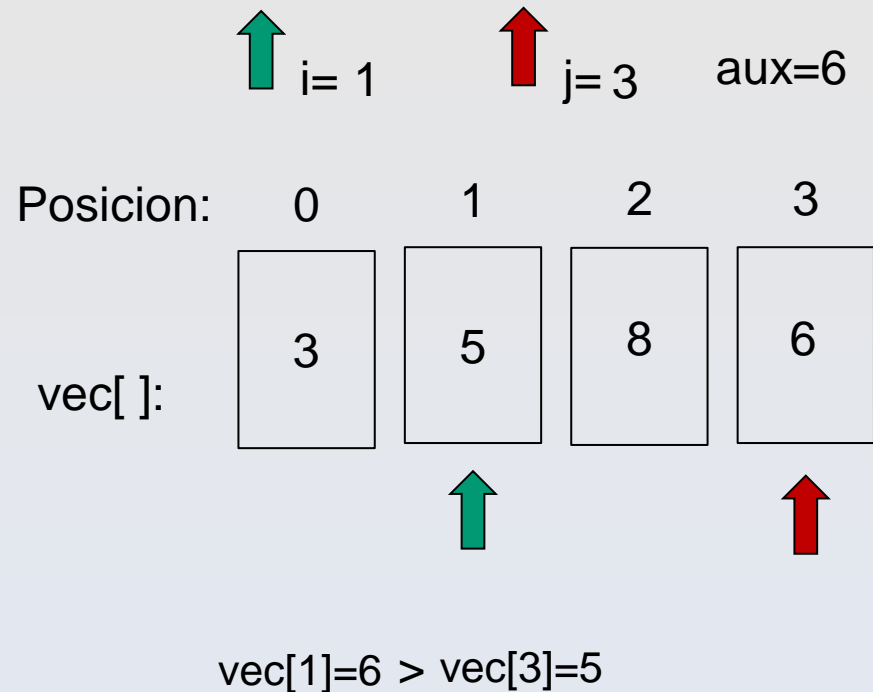


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

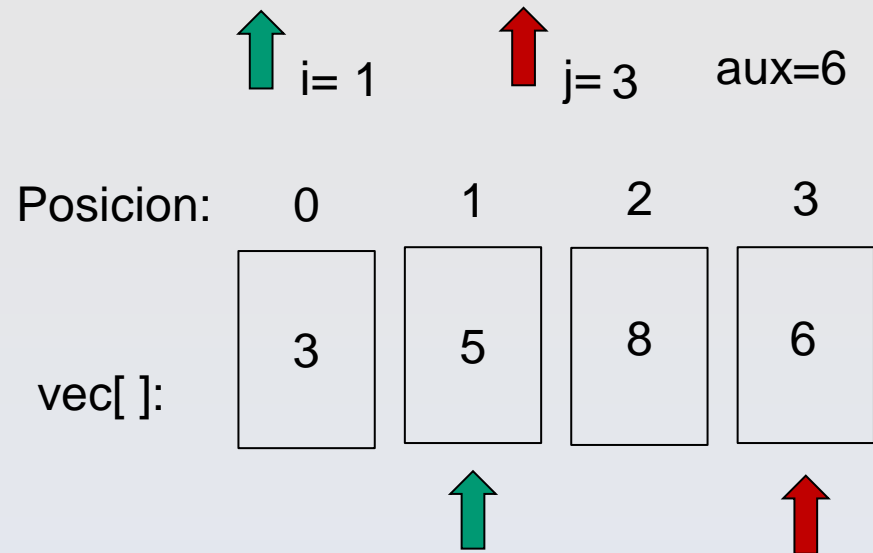


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

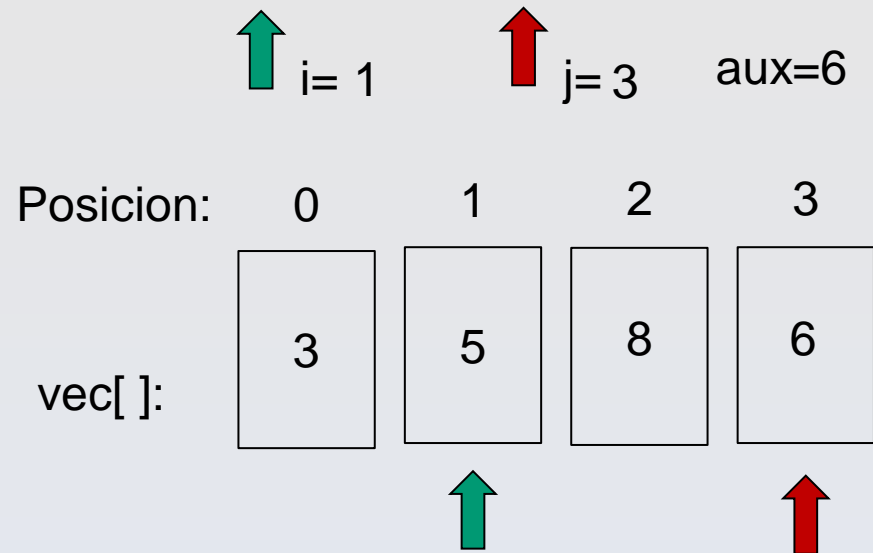


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
for(i=0;i<FIL-1;i++)  
  { for(j=i+1;j<FIL;j++)  
    { if(vec[i]>vec[j])  
      { aux=vec[i];  
        vec[i]=vec[j];  
        vec[j]=aux;}}}
```

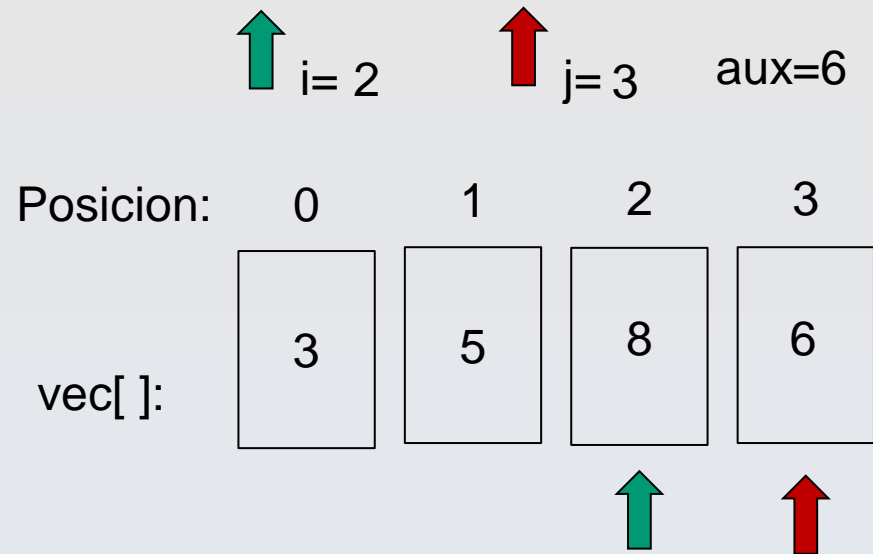


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

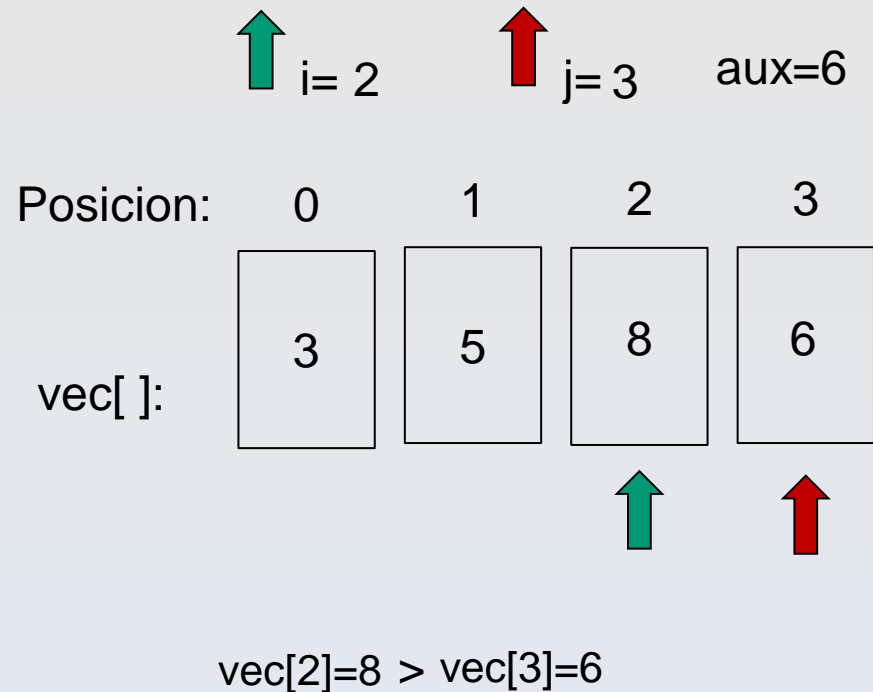


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        → { if(vec[i]>vec[j])  
                { aux=vec[i];  
                  vec[i]=vec[j];  
                  vec[j]=aux; } } }
```

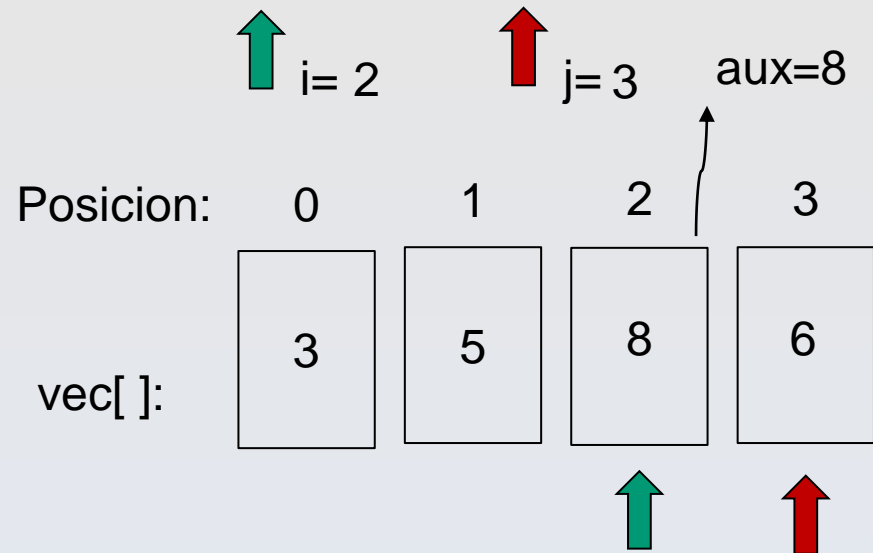


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            → { aux=vec[i];  
                vec[i]=vec[j];  
                vec[j]=aux; } } }
```



$vec[2]=8 > vec[3]=6$

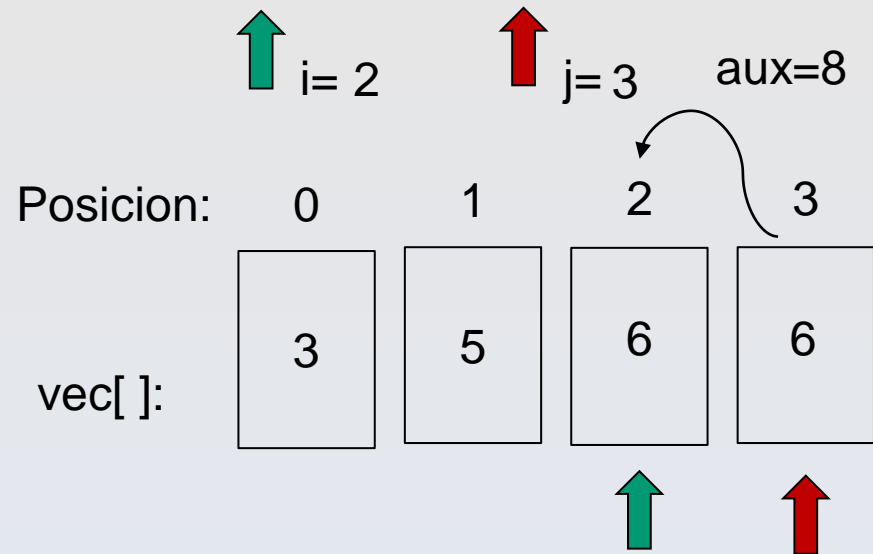
$aux=8$

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              → vec[i]=vec[j];  
                vec[j]=aux; } } }
```



$vec[2]=8 > vec[3]=6$

$aux=8$

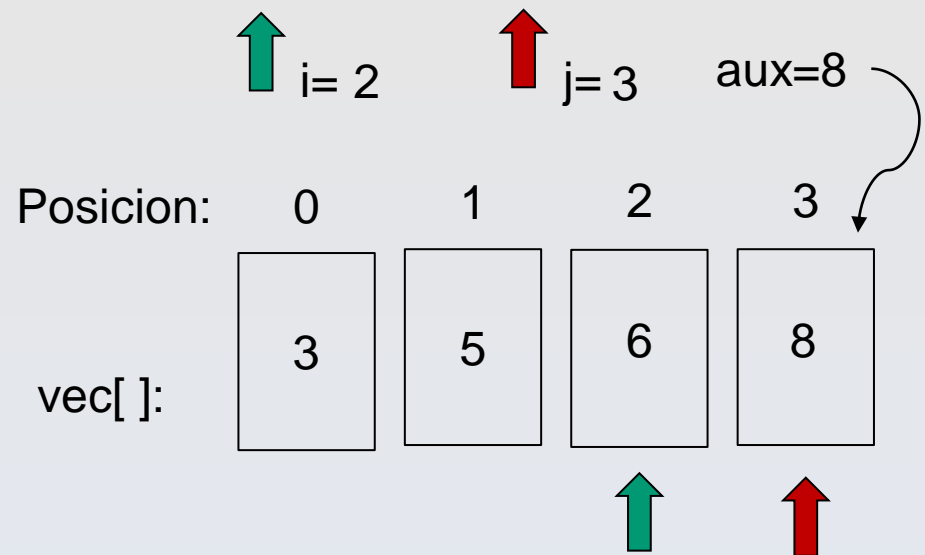
$vec[2]=6$

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              → vec[j]=aux; } } }
```



$vec[2]=8 > vec[3]=6$

$aux=8$

$vec[2]=6$

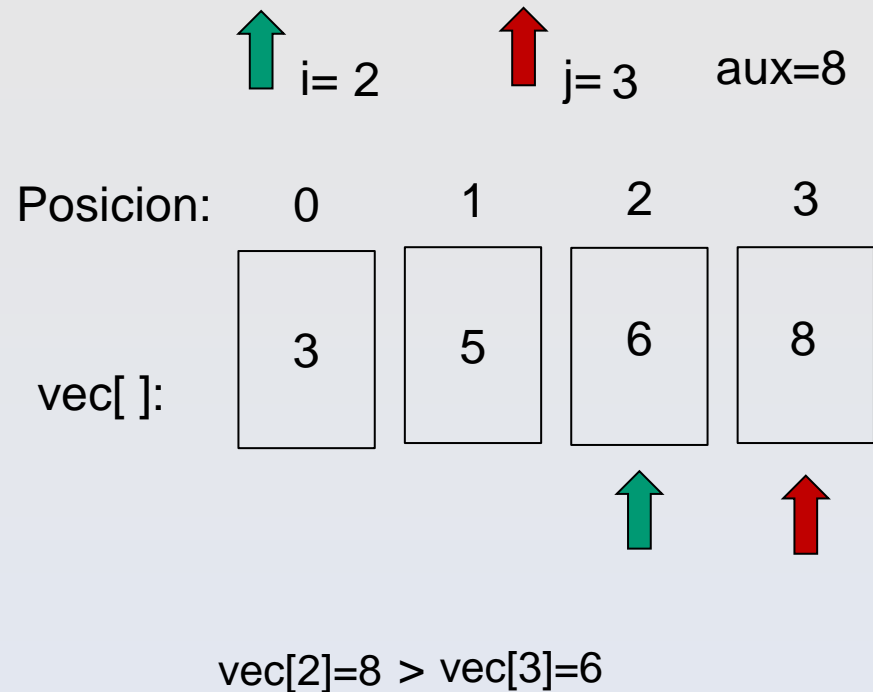
$vec[3]=8$

CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    → { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

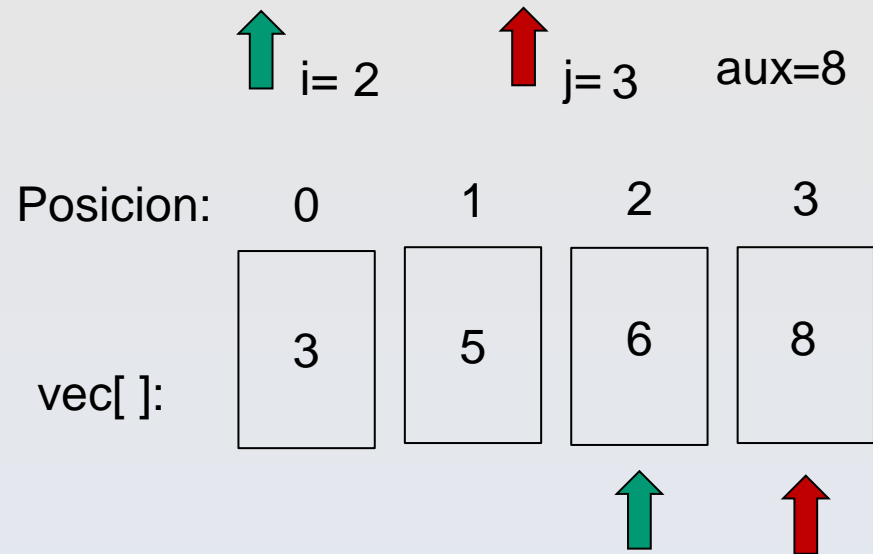


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
→ for(i=0; i<FIL-1; i++)  
    { for(j=i+1; j<FIL; j++)  
        { if(vec[i]>vec[j])  
            { aux=vec[i];  
              vec[i]=vec[j];  
              vec[j]=aux; } } }
```

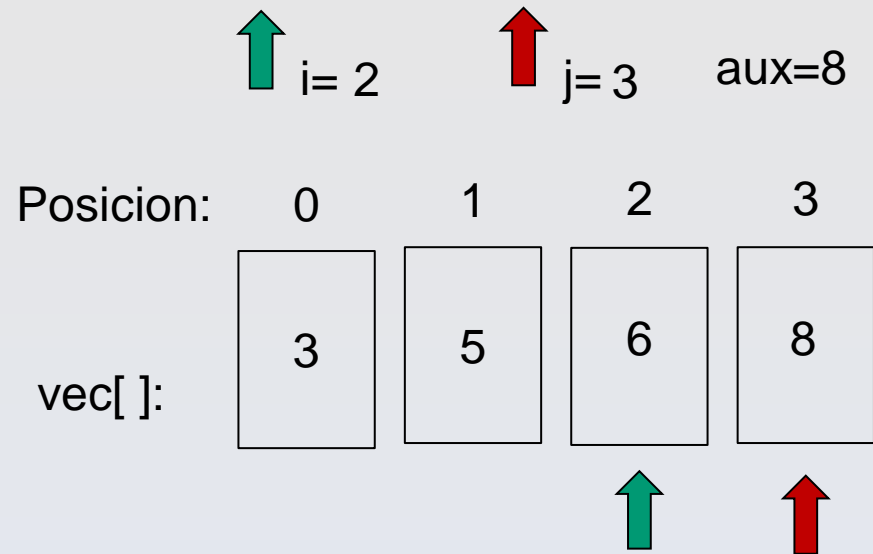


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
for(i=0;i<FIL-1;i++)  
    { for(j=i+1;j<FIL;j++)  
        {if(vec[i]>vec[j])  
            {aux=vec[i];  
            vec[i]=vec[j];  
            vec[j]=aux;}}}
```


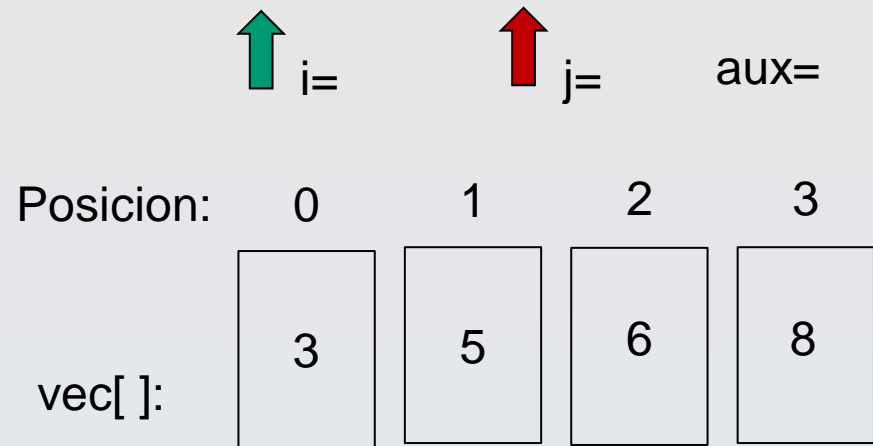


CLASE 5

Arrays: Búsqueda

Método de burbujeo

```
for(i=0;i<FIL-1;i++)  
{ for(j=i+1;j<FIL;j++)  
  {if(vec[i]>vec[j])  
    {aux=vec[i];  
    vec[i]=vec[j];  
    vec[j]=aux;}}}
```



Array Ordenado
de manera
ascendente

