CICLO 01

[FORMACIÓN POR CICLOS]

Fundamentos de Programación

Colección de datos

Borrado en vector

Ingeni@ Soluciones TIC

UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

Operación de borrado en vector

d = 16

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
V	14	3	1	6	2	8	4	9	16	28	5	39	44	7	12										

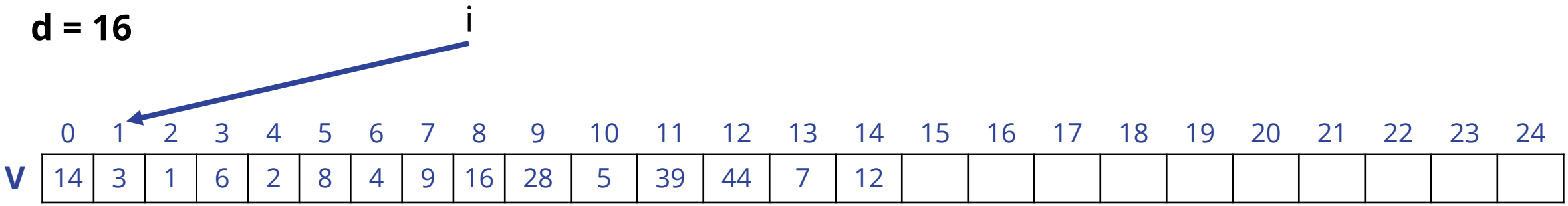
El proceso consiste en buscar en cuál posición está el dato a borrar.

Luego se desplazan, una posición hacia la izquierda, los datos que están a continuación del dato a borrar con el fin de que no queden espacios intermedios libres en el vector.

Por último se actualiza **V[0]**.

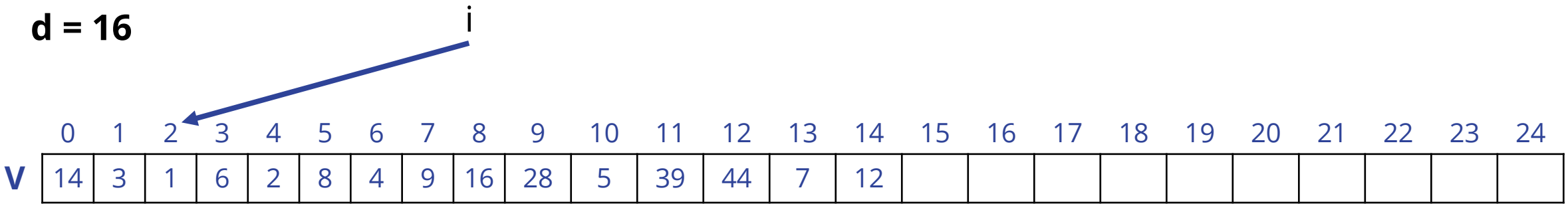
Tendremos dos subprogramas: uno que busca el dato a borrar (este subprograma retorna la posición donde se halla el dato a borrar) y otro que efectúa el movimiento de datos en el vector.

Es importante notar que en el subprograma para buscar el dato a borrar, nuestro algoritmo retornará **–1** si no encuentra el dato buscado.



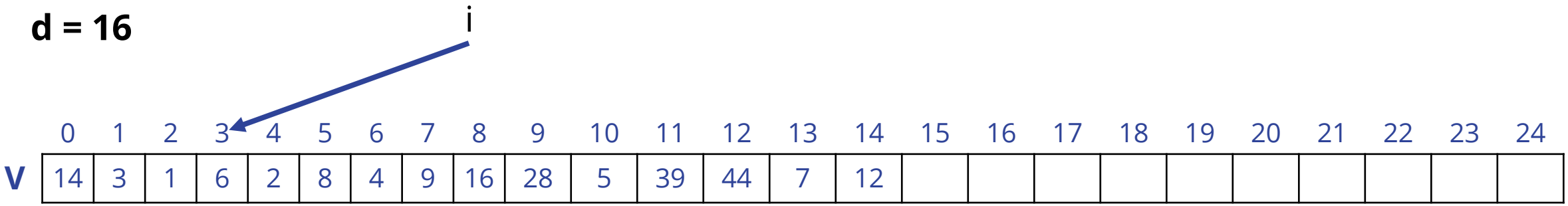
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



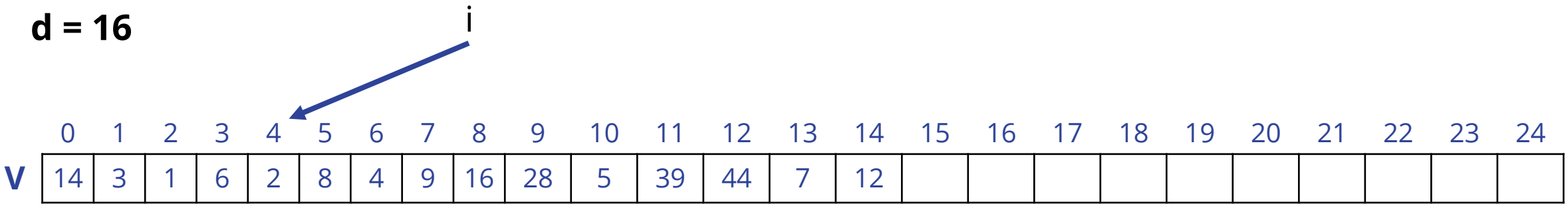
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



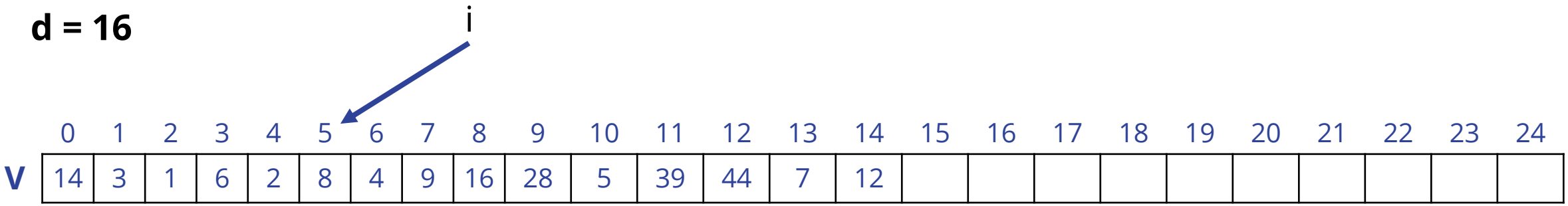
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



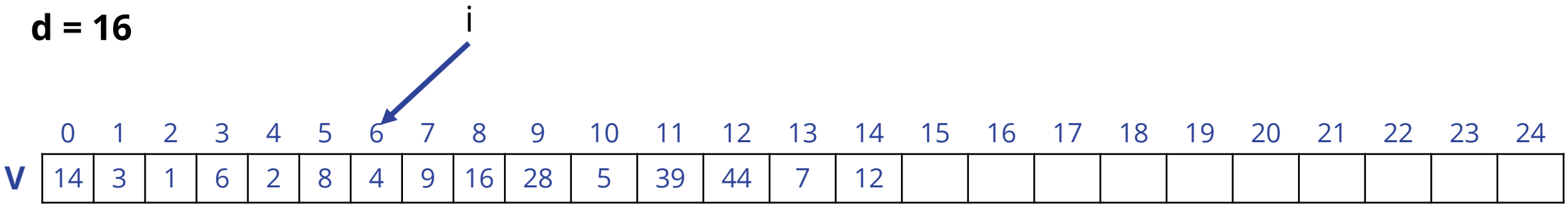
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



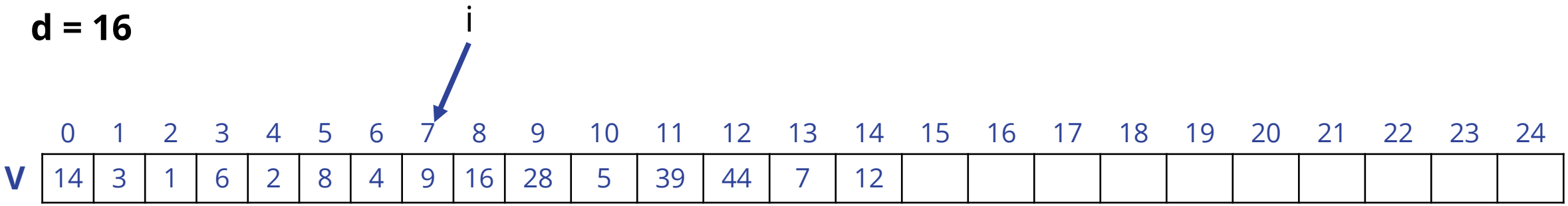
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



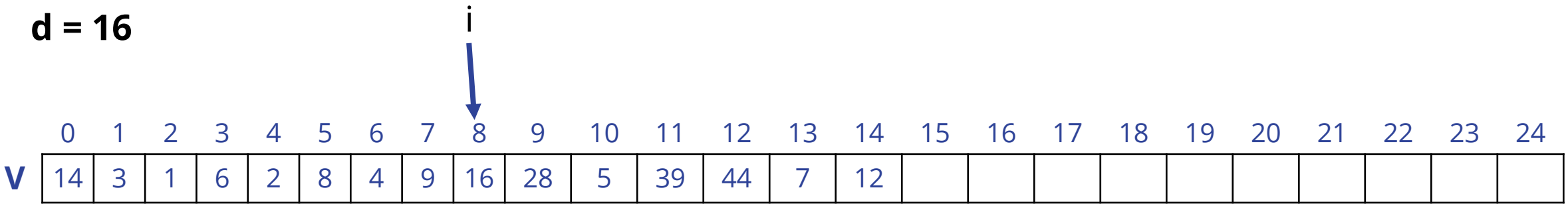
Nuestros subprogramas son:

```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



Nuestros subprogramas son:

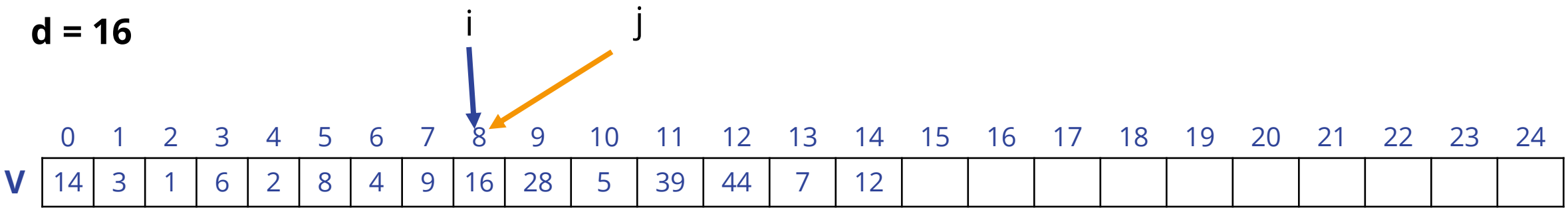
```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```



Nuestros subprogramas son:

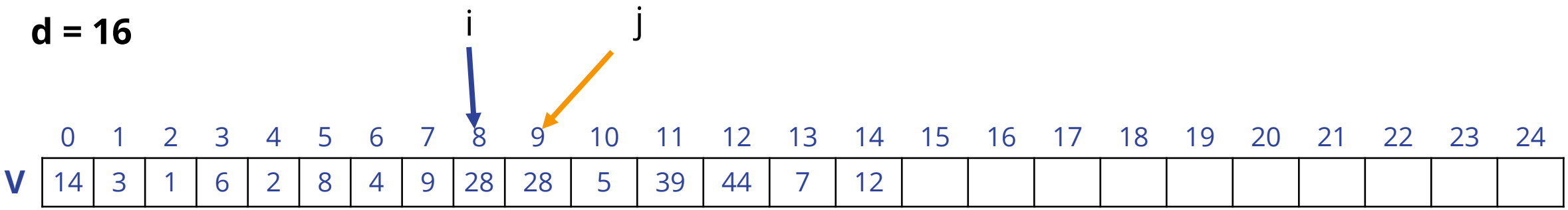
```
def  buscarDato(V, d):
        i = 1
        while  i <= V[0]  and  V[i] != d:
                i = i + 1
        if i <= V[0]:
                return  i
        return  −1
```

d = 16



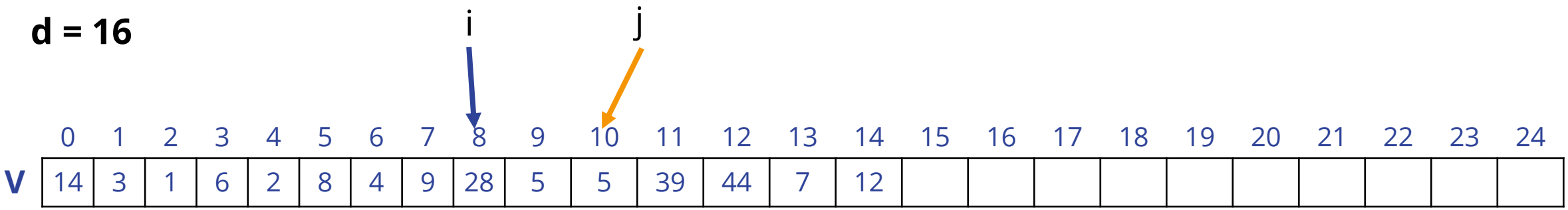
Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



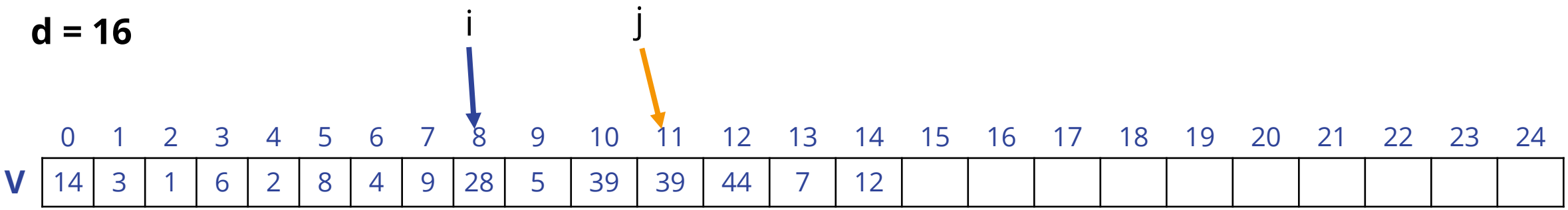
Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



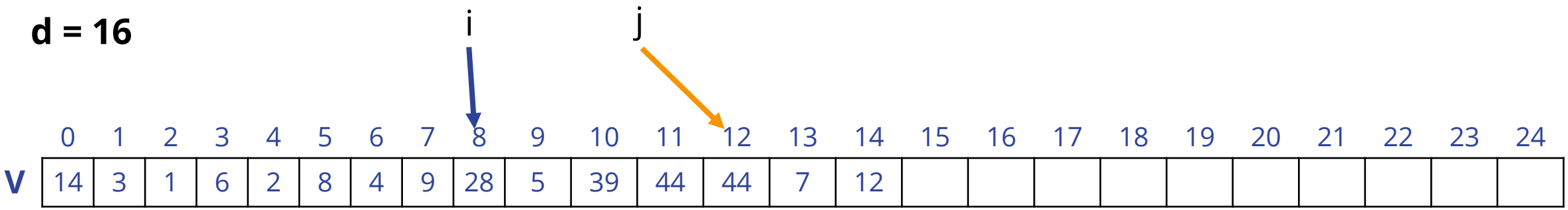
Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



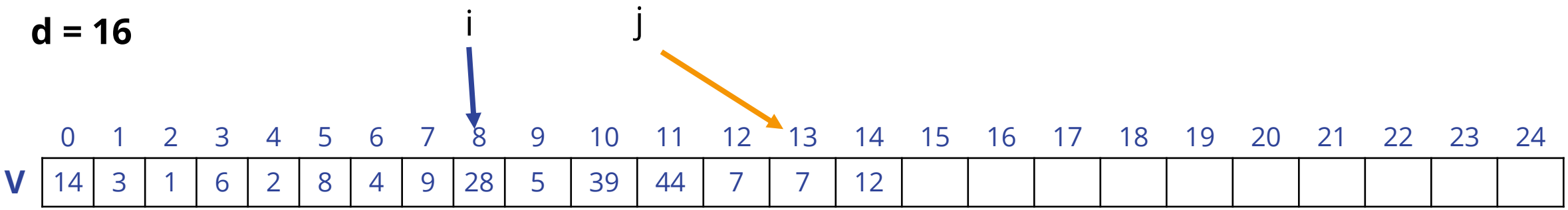
Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



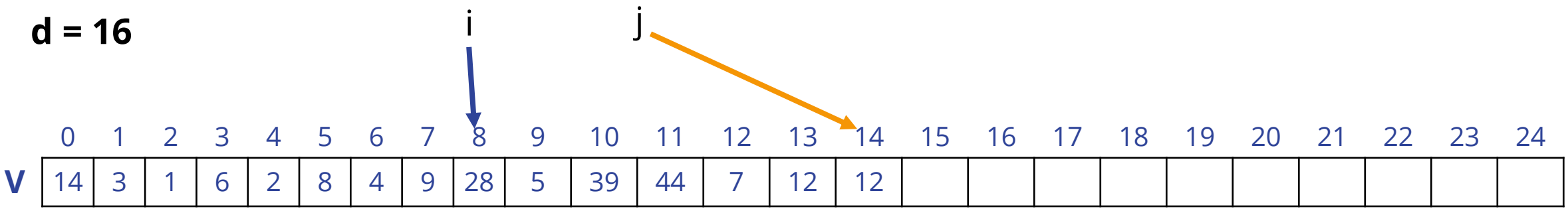
Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



Nuestros subprogramas son:

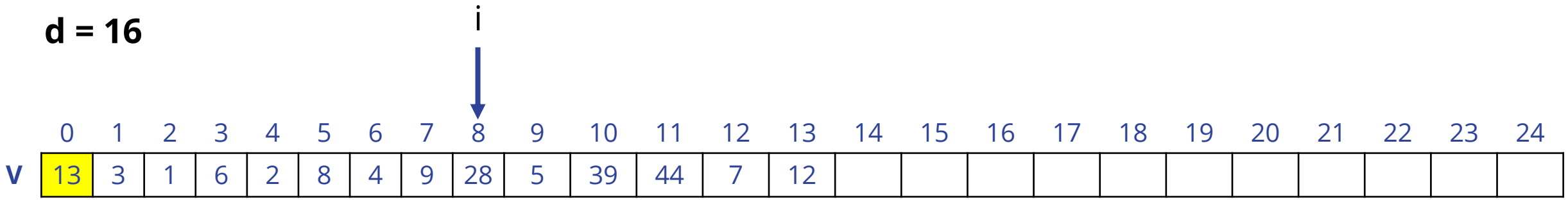
```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```



Nuestros subprogramas son:

```
def  borrar(V, i):
        for  j  in  range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```

d = 16




```
def borrar(V, i):
        for j in range(i, V[0]):
                V[j] = V[j + 1]
        V[0] = V[0] – 1
```