

**PROGRAMACIÓN I**

AÑO 2025

# **ESTRUCTURA DE DATOS ARREGLO. ORDENACIÓN EN VECTORES**

# Temas

- ✓ CONCEPTO DE ORDENACIÓN.
- ✓ MÉTODOS DE ORDENACIÓN:
  - ✓ Selección
  - ✓ Intercambio

# MÉTODOS PARA ORDENAR VECTORES



# **MÉTODOS DE ORDENACIÓN EN VECTORES**

**El proceso por el cual, un grupo de elementos puede ser ordenado se conoce como algoritmo de ordenación.**

**¿Por qué es importante una operación de ordenación en arreglos?**

# MÉTODO DE SELECCIÓN

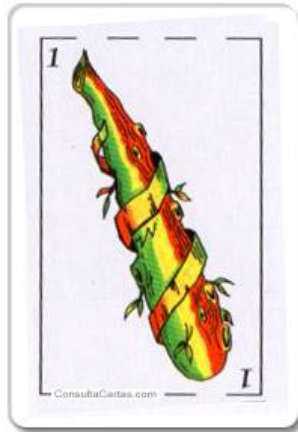


# MÉTODO DE SELECCIÓN (PASADA 1)

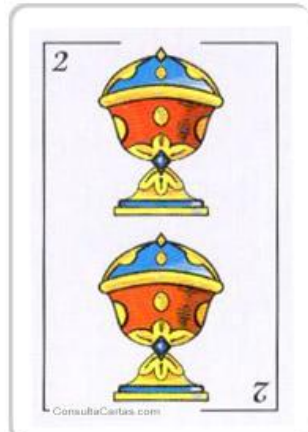
¿Quién es el menor  
de mi derecha?



Posición 1



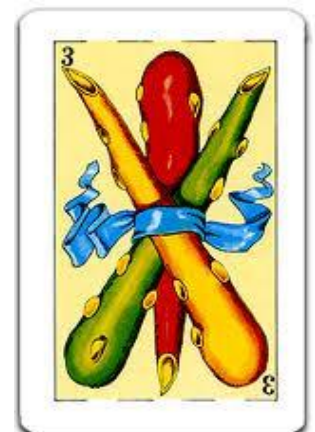
Posición 2



Posición 3



Posición 4



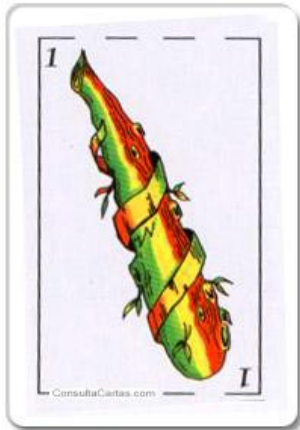
Posición 5

Notar que el menor queda en la  
Primera posición al terminar  
la 1era pasada

Auxiliar

# MÉTODO DE SELECCIÓN (PASADA 2)

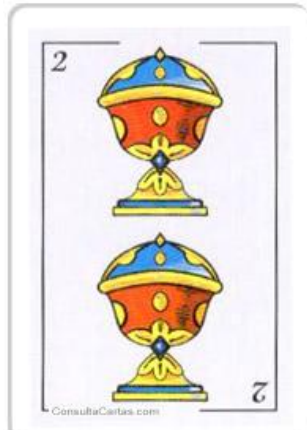
¿Quién es el menor  
de mi derecha?



Posición 1



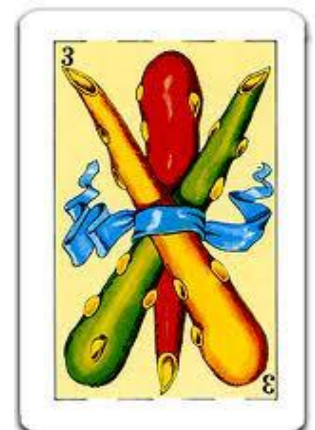
Posición 2



Posición 3



Posición 4



Posición 5

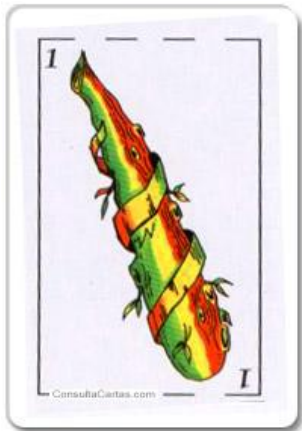
Notar que el 2do mínimo queda en la  
segunda posición al terminar  
la 2da pasada

Auxiliar

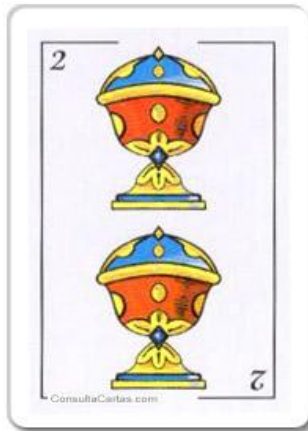


# MÉTODO DE SELECCIÓN (PASADA 3)

¿Quién es el menor  
de mi derecha?



Posición 1



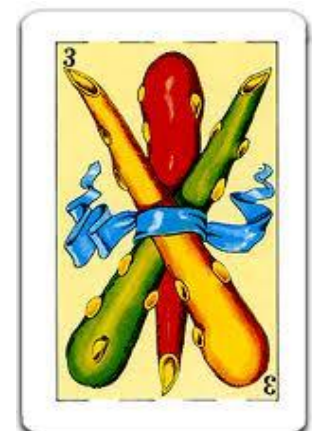
Posición 2



Posición 3



Posición 4



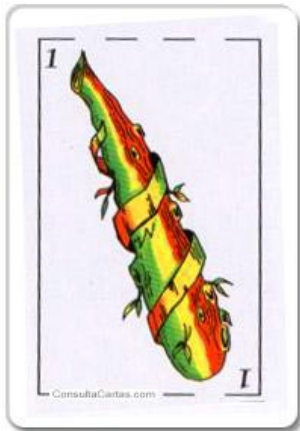
Posición 5

Notar que el 3er mínimo queda en la  
Tercera posición al terminar  
la 3era pasada

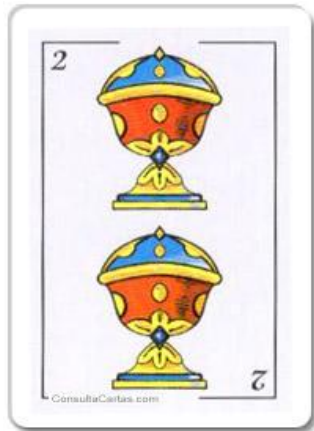
Auxiliar

# MÉTODO DE SELECCIÓN (PASADA 4)

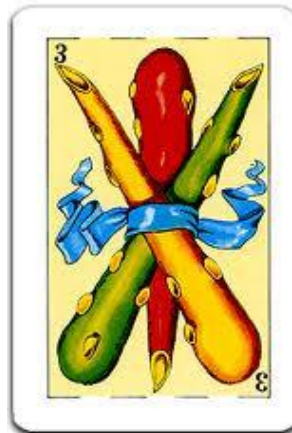
¿Quién es el menor  
de mi derecha?



Posición 1



Posición 2



Posición 3



Posición 4



Posición 5

Notar que el 4to mínimo queda en la  
Cuarta posición al terminar  
la 4ta pasada

Auxiliar

# MÉTODOS DE ORDENACIÓN EN VECTORES

Para los algoritmos vamos a tener en cuenta las siguientes declaraciones:

**Const** maxlen = 100; *{máxima longitud del vector}*

**Type**

TipoElem = integer; *{ tipo de datos del vector }*

Indice = 0.. maxlen ;

vector = **Array** [ 1..maxlen] **of** TipoElem;

vector\_dimL = record

    v: vector;

    dimL: indice;

end;

## **Método de Selección**

$\Rightarrow$  en cada pasada se elige un mínimo y se lo ubica en el lugar que le corresponde.

$\Rightarrow$   $n-1$  pasadas

**Veamos un ejemplo:**

## Método de Selección

8	2	1	6
---	---	---	---

La primera vez vamos a poner el mínimo en la posición 1

Luego, buscamos el mínimo a partir de la segunda posición y lo ubicamos allí.

Así siguiendo...

p	i		
8	2	1	6

## Método de Selección

**Procedure** Ordenar\_seleccion ( var vD: vector\_dimL );

**var** i, j, pos: indice;

aux : tipoElem;

**begin**

**for** i:= v1 **to** vD.dimL-1 **do begin**

*{ busca el mínimo v[p] entre v[i], ..., v[N] }*

pos:= i;

**for** j := i+1 **to** vD.dimL **do**

**if** (vD.v[ j ] < vD.v[ pos ]) **then** p:=j;

*{intercambia v[i] y v[pos] }*

aux := vD.v[ pos ];

vD.v[ pos ] := vD.v[ i ];

vD.v[ i ] := aux;

**end;**

**end;**

l = 1 a 3

J= i+1 a 4

Item=1

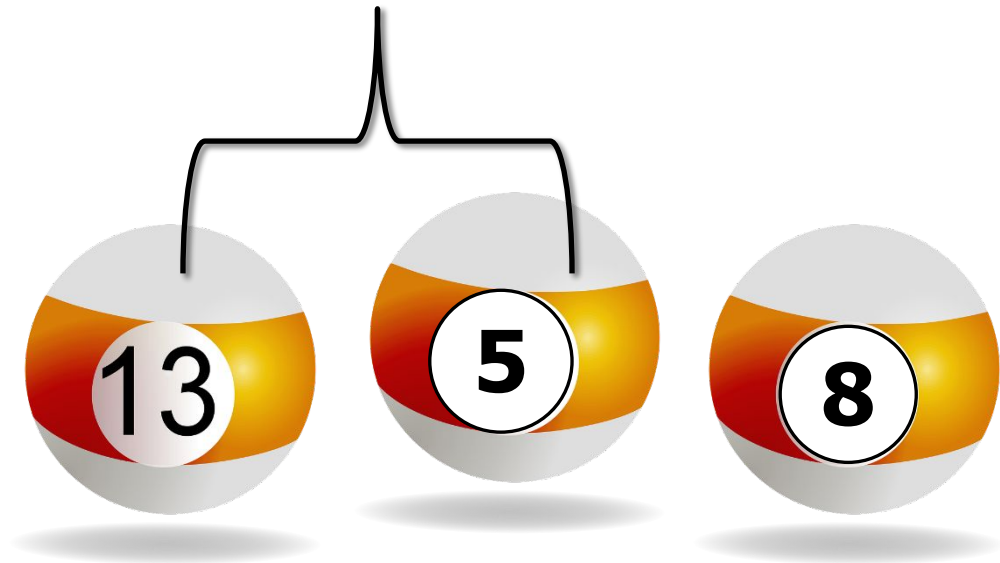
V[3]=8

V[1]=1

l= 2

# MÉTODO DE INTERCAMBIO

Comparo adyacentes



# MÉTODO DE INTERCAMBIO

Se comparan e intercambian datos adyacentes  
 $\Rightarrow n-1$  pasadas

Al fin de cada pasada **el mayor viaja y queda al fondo del arreglo** ( no es necesario re-compararlo )

Veamos el ejemplo:

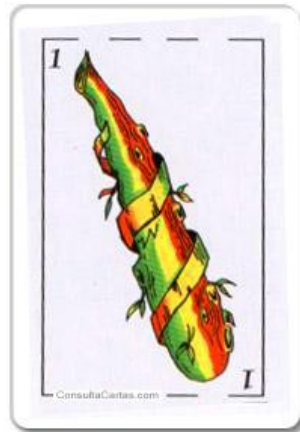


# Método de Intercambio (pasada 1)

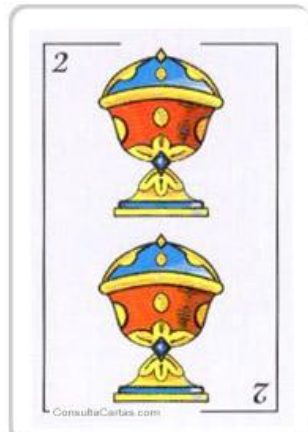
¿Cómo soy respecto de mi vecino?



Posición 1



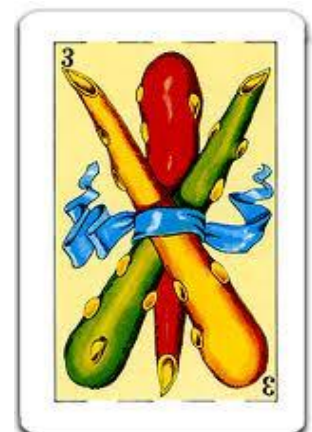
Posición 2



Posición 3



Posición 4

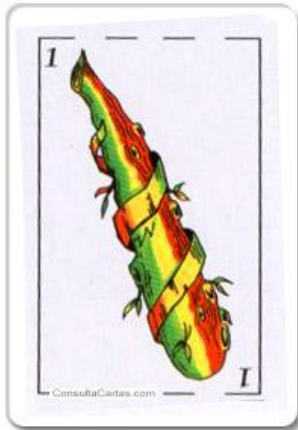


Posición 5

Auxiliar

# Método de Intercambio (pasada 1)

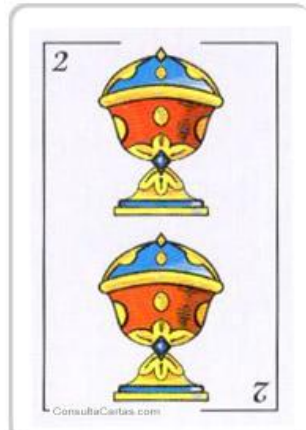
¿Cómo soy respecto de mi vecino?



Posición 1



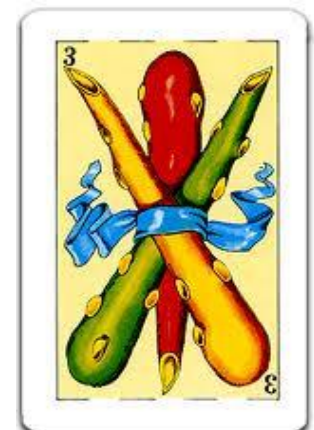
Posición 2



Posición 3



Posición 4



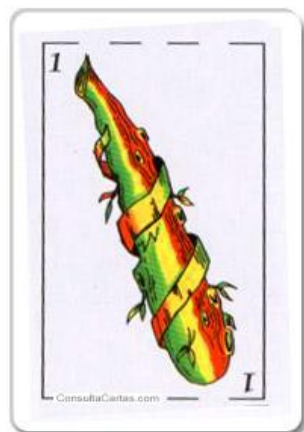
Posición 5

Auxiliar

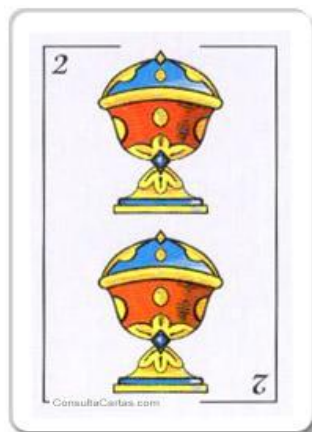
# Método de Intercambio (pasada 1)

¿Cómo soy respecto  
de mi vecino?

¿Cómo soy respecto de  
mi vecino?



Posición 1



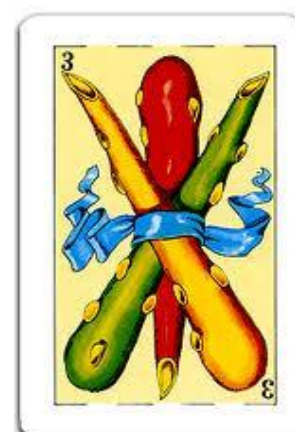
Posición 2



Posición 3



Posición 4



Posición 5

Notar que al terminar la 1era pasada,  
el mayor está ubicado al final.  
No es necesario volver a compararlo

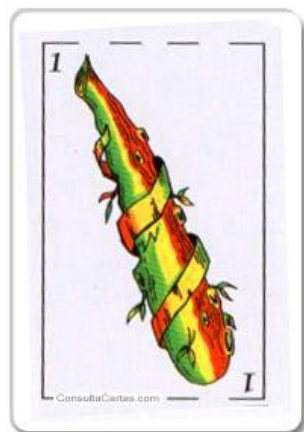
Auxiliar

# Método de Intercambio (pasada 2)

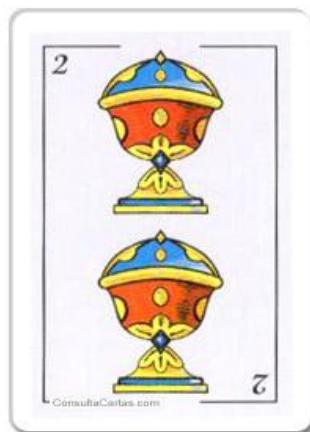
¿Cómo soy respecto  
mi vecino?

¿Cómo soy re  
mi vecino?

¿Cómo soy respecto de  
mi vecino?



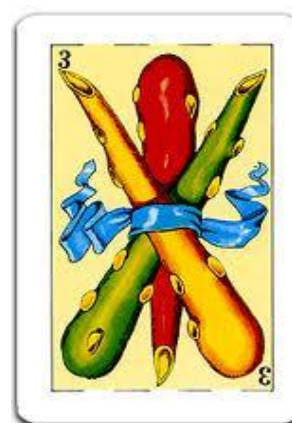
Posición 1



Posición 2



Posición 3



Posición 4



Posición 5

¿Hasta dónde se recorre el arreglo?  
¿Es necesario comparar con el último?

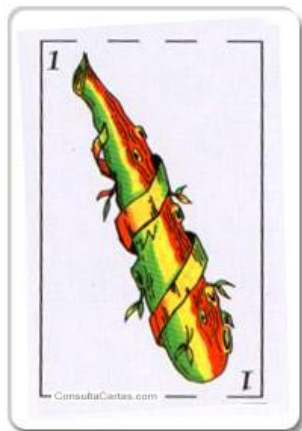
Auxiliar

# Método de Intercambio (pasada 3)

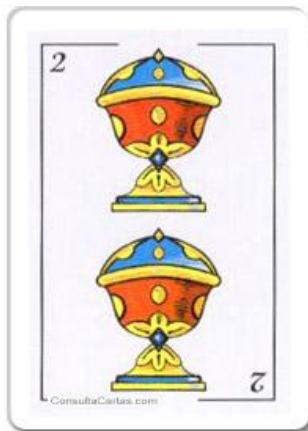
¿Cómo soy respecto  
de mi vecino?

¿Cómo soy respecto  
de mi vecino?

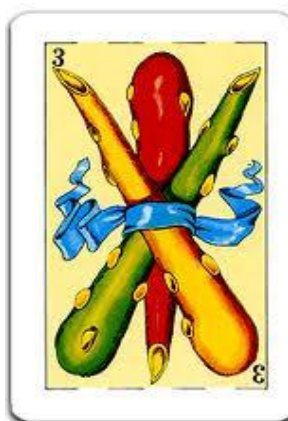
¿Cómo soy respecto  
de mi vecino?



Posición  
1



Posición  
2



Posición  
3



Posición  
4



Posición  
5

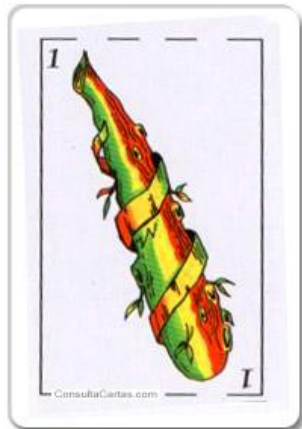
Notar que al terminar  
la 3era pasada,  
los últimos tres están  
ordenados

Auxiliar



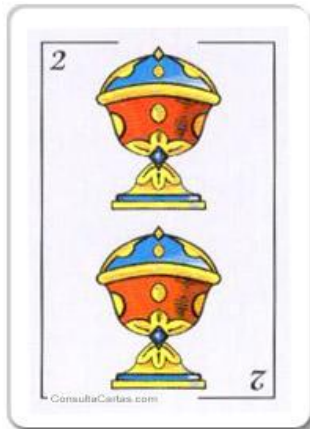
# Método de Intercambio (pasada 4)

¿Cómo soy respecto de mi vecino?



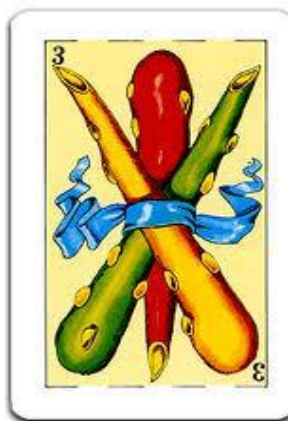
Posición

1



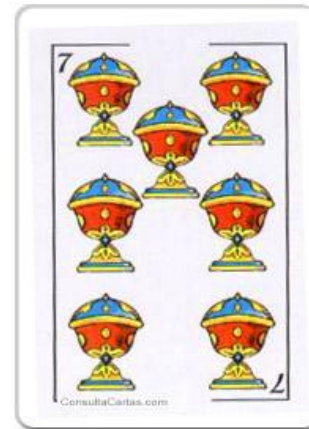
Posición

2



Posición

3



Posición

4



Posición

5

Auxiliar

**Procedure** Ordenar\_intercambio ( var vD: vector\_dimL );

**var** i, j: indice;

aux : tipoElem;

**begin**

**for** i:= vD.dimL **downto** 2 **do begin**

*{ poner el mayor elemento de  $v[1], \dots, v[i]$  en  $v[i]$  }*

**for** j := 1 **to** i-1 **do**

**if** (vD.v[ j ] > vD.v[ j+1 ] )

**then begin** *{intercambiar elementos}*

aux := vD.v[ j ];

vD.v[ j ] := vD.v[ j+1 ];

vD.v[ j+1 ] := aux;

**end;**

**end;**

**end;**

## **Método de Intercambio con centinela**

Podemos mejorar la eficiencia del método haciendo uso de una variable lógica o centinela

**==>** si no hay intercambio, el arreglo ya está ordenado

**Veamos el ejemplo:**

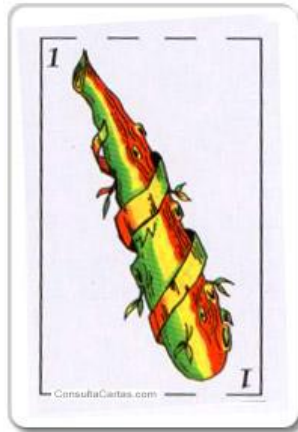


# Método de Intercambio (pasada 1)

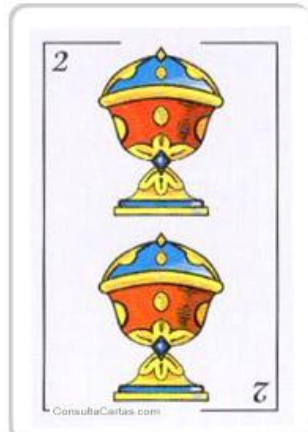
¿Cómo soy respecto de mi vecino?



Posición  
1



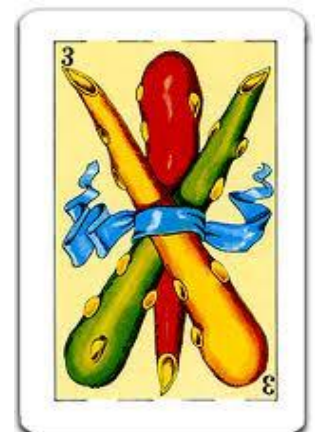
Posición  
2



Posición  
3



Posición  
4

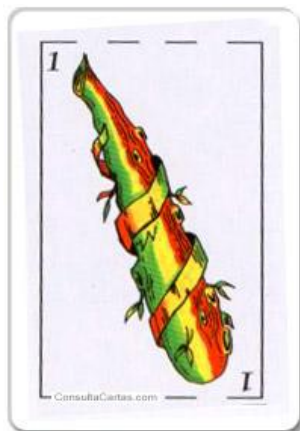


Posición  
5

Auxiliar

# Método de Intercambio (pasada 1)

¿Cómo soy respecto  
de mi vecino?



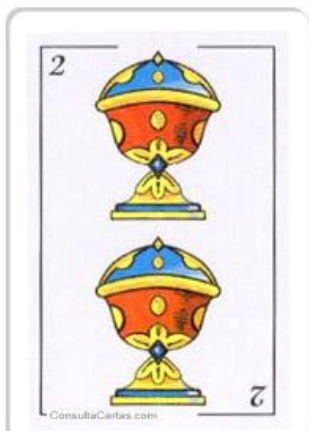
Posición

1



Posición

2



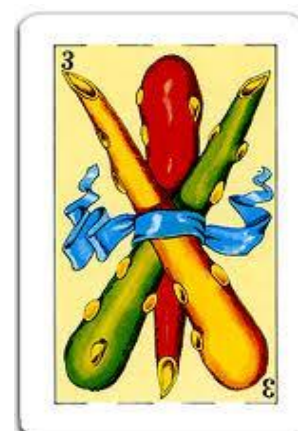
Posición

3



Posición

4



Posición

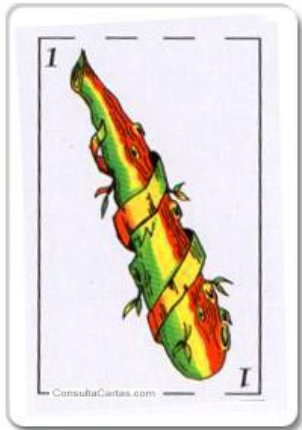
5

Auxiliar

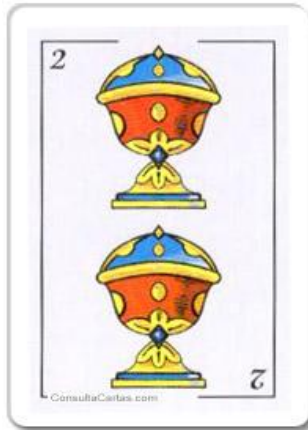
# Método de Intercambio (pasada 1)

¿Cómo soy respecto  
de mi vecino?

¿Cómo soy respecto  
de mi vecino?



Posición  
1



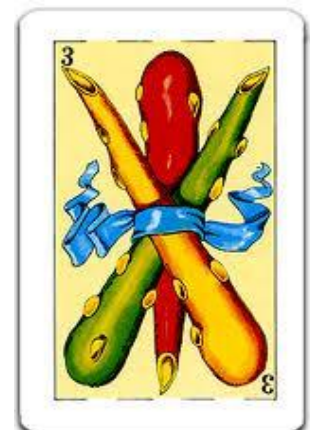
Posición  
2



Posición  
3



Posición  
4



Posición  
5

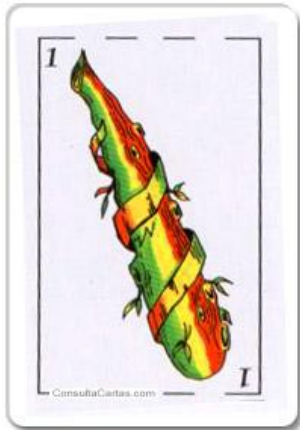
Auxiliar

# Método de Intercambio (pasada 2)

¿Cómo soy respecto  
de mi vecino?

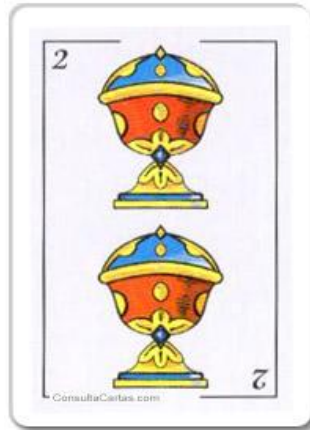
¿Cómo soy  
de mi vecino?

¿Cómo soy respecto  
de mi vecino?



Posición

1



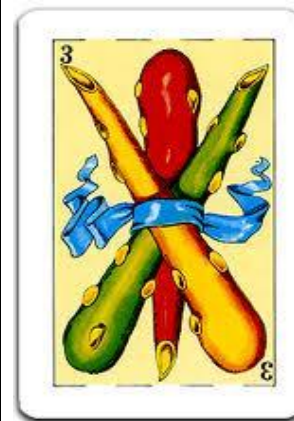
Posición

2



Posición

3



Posición

4



Posición

5

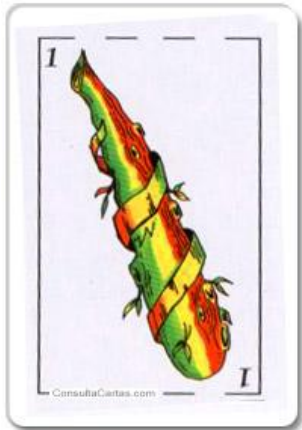
¿Habrá más cambios en las  
próximas pasadas?  
¿Cómo se puede determinar que no  
habrá mas cambios?

Auxiliar

# Método de Intercambio (pasada 3)

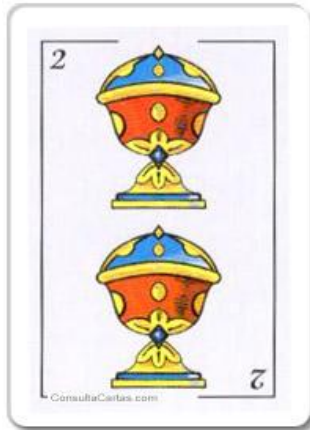
¿Cómo soy respecto  
de mi vecino?

¿Cómo soy respecto  
de mi vecino?



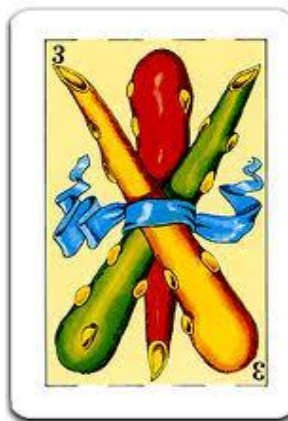
Posición

1



Posición

2



Posición

3



Posición

4



Posición

5

Cómo no hubo ningún  
cambio, se asegura que  
el vector está ordenado

Auxiliar



```
Procedure Ordenar_intercambio_centinela ( var vD: vector_dimL );  
var i, j: indice;  
    aux : tipoElem; huboCambio: boolean;  
begin  
    i:= vD.dimL;  
    repeat  
        huboCambio:= false;  
        for j := 1 to i-1 do  
            if (vD.v[ j ] > vD.v[ j+1 ] )  
                then begin {intercambiar elementos}  
                    aux := vD.v[ j ];  
                    vD.v[ j ] := vD.v[ j+1 ];  
                    vD.v[ j+1 ] := aux;  
                    huboCambio := True;  
                end;  
        i := i - 1;  
    until ( (i < 2) or (HuboCambio = false));  
end;
```