
INTERSECCIÓN

ANÁLISIS

1. PROBLEMA

Crear un arreglo con los elementos que se repiten en dos arreglos dados. El resultado se guardará en el tercer arreglo dado.

2. DATOS DE ENTRADA

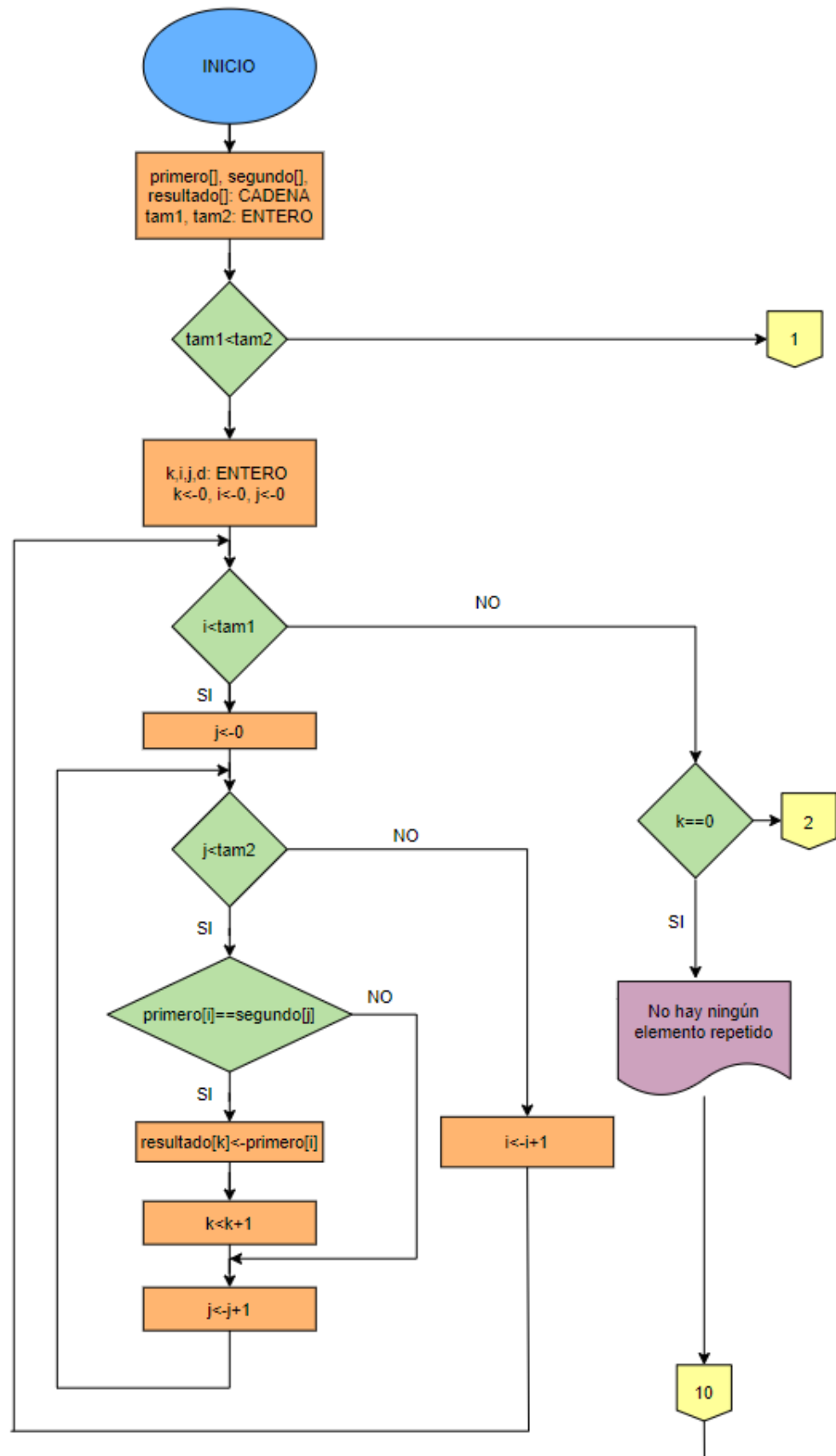
Tres arreglos de tipo entero

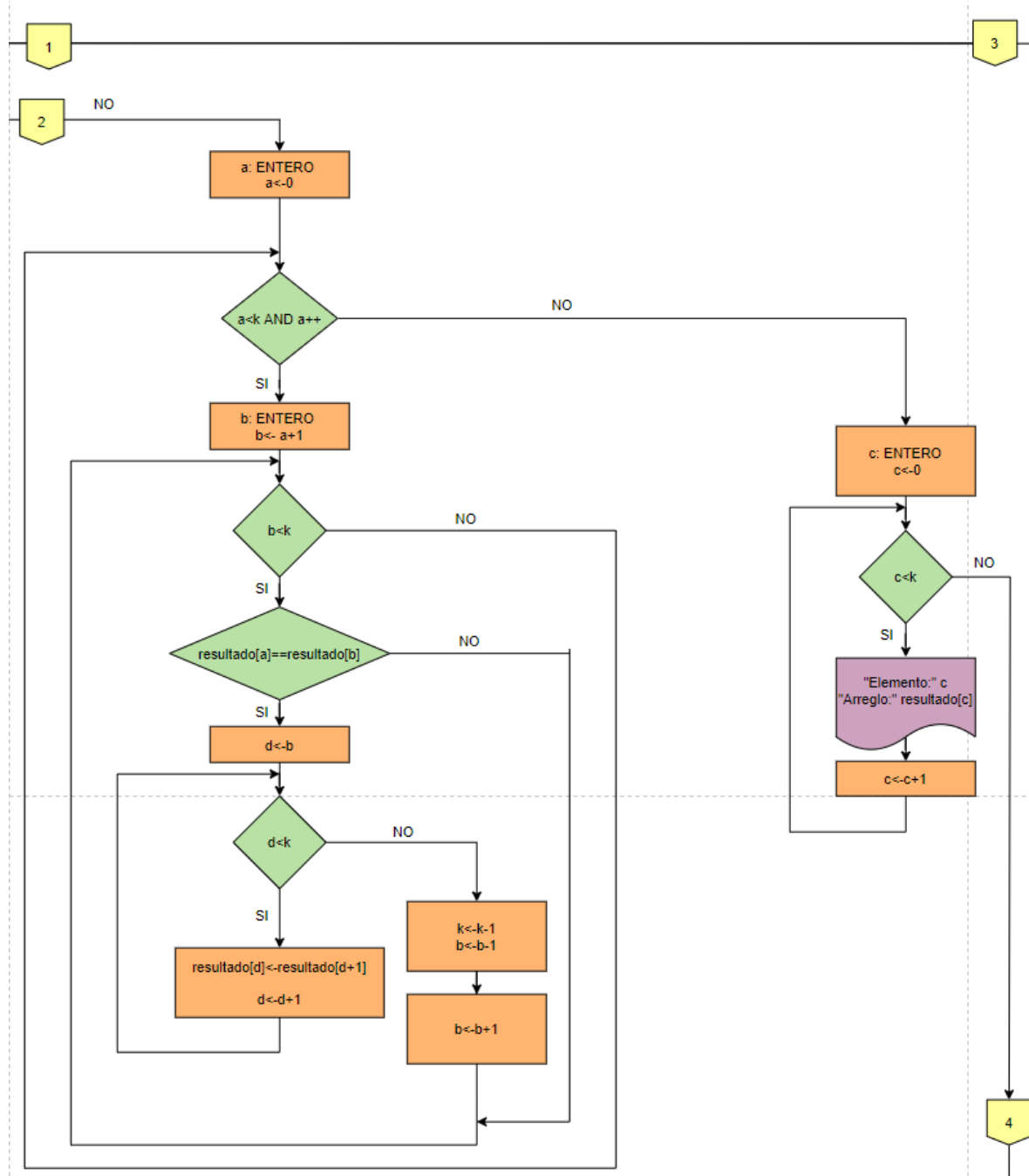
3. DATOS DE SALIDA

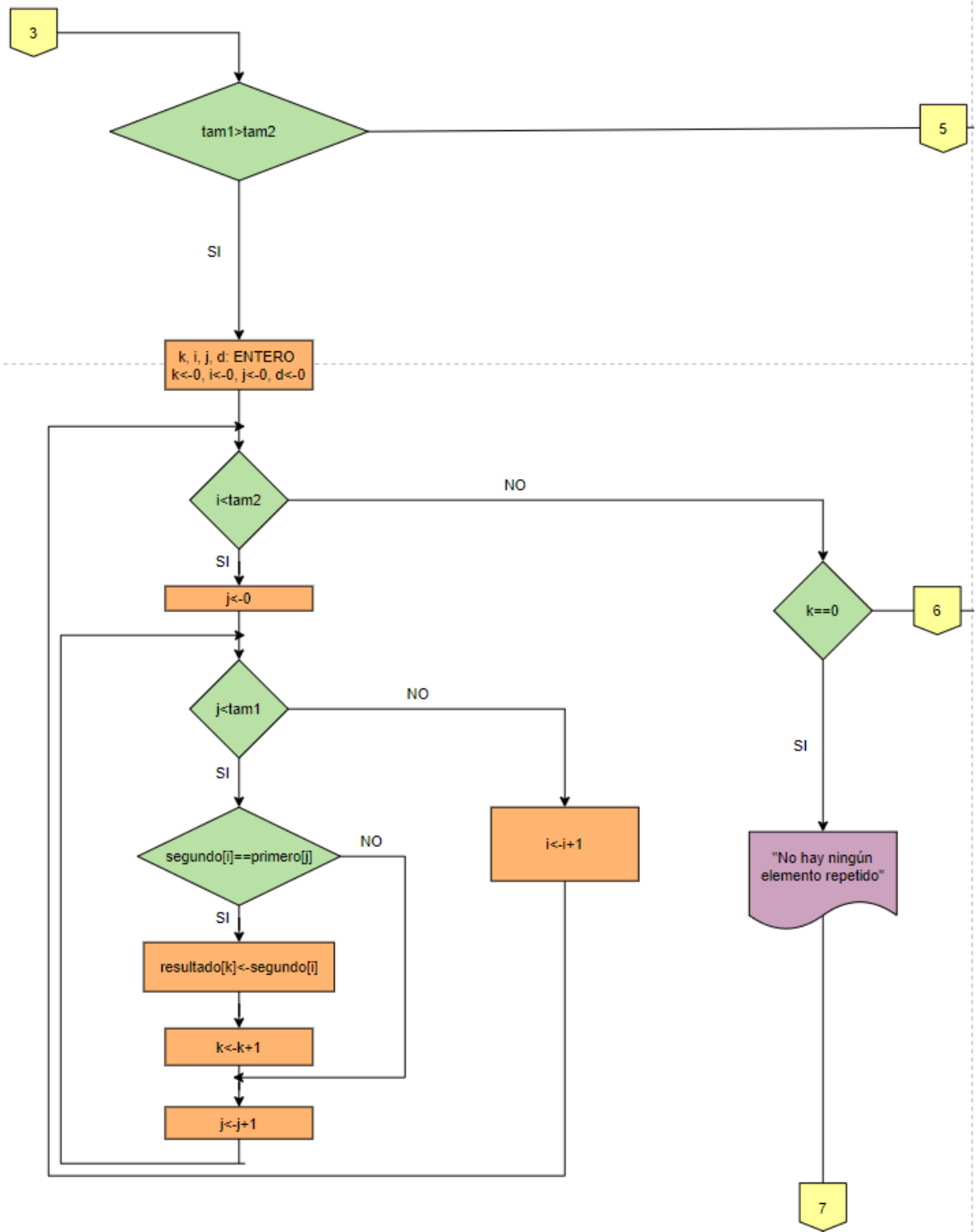
Un arreglo de tipo entero hecho con los elementos repetidos de los 2 anteriores arreglos.

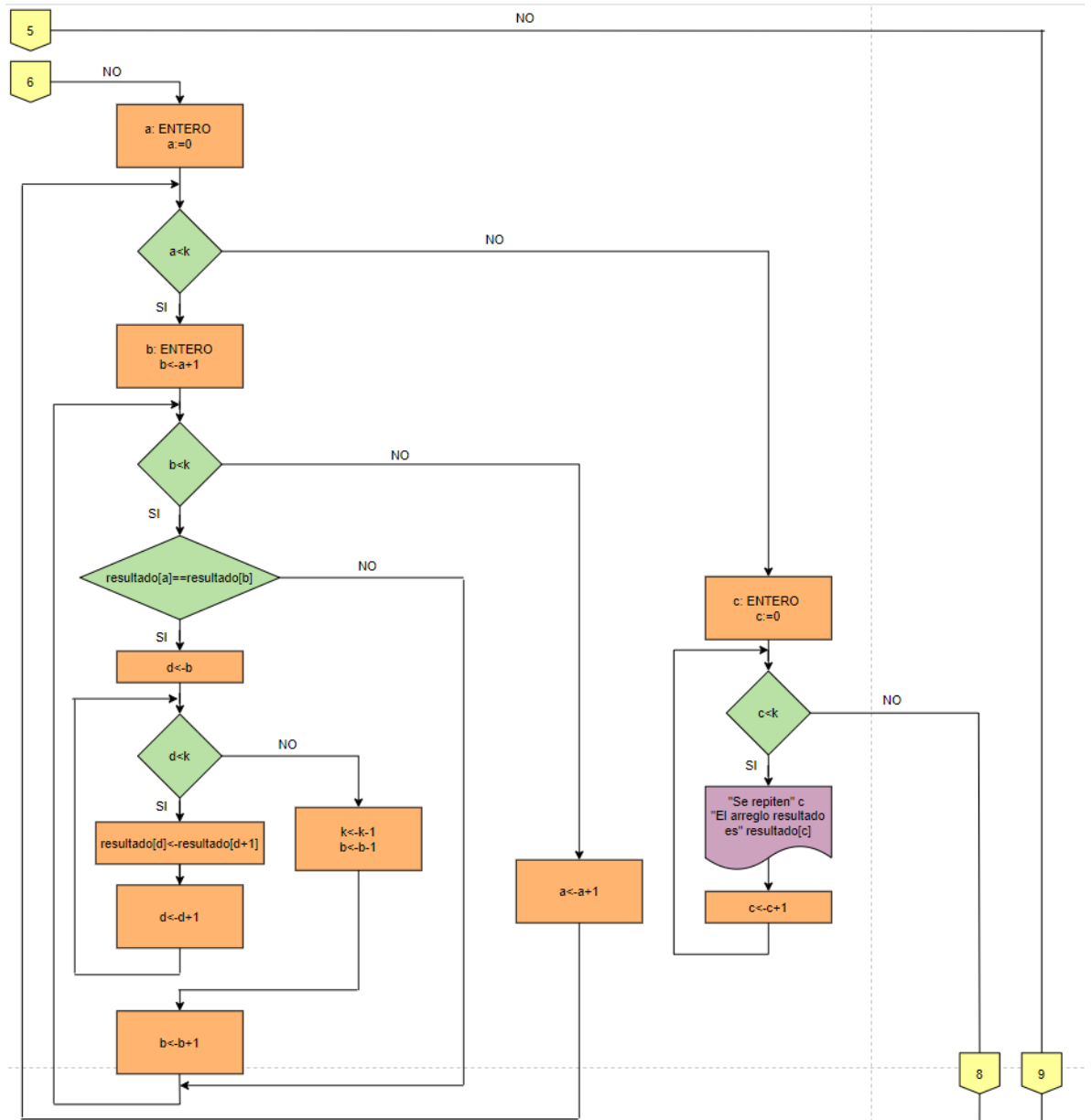
DISEÑO

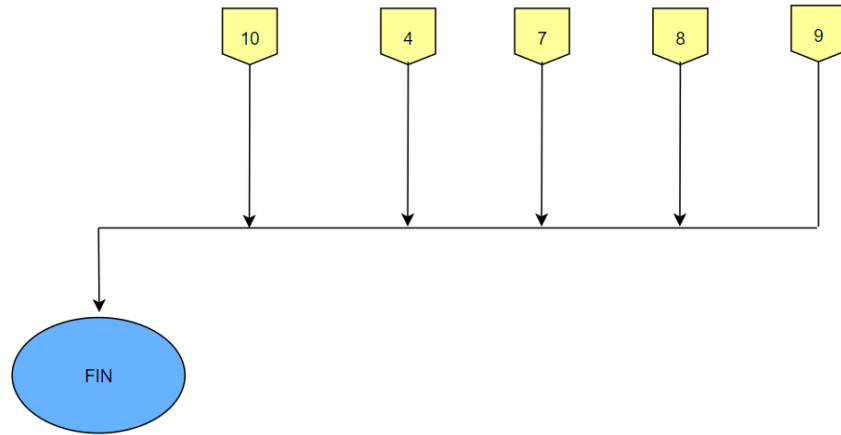
4. DIAGRAMA DE FLUJO











5. PSEUDOCODIGO

INICIO

primero:ENTERO, tam1:ENTERO, segundo:ENTERO, tam2:ENTERO,
resultados:ENTERO

FUNC ENCONTRAR

SI (tam1<tam2) ENTONCES

i=0:ENTERO, j=0:ENTERO, k=0:ENTERO, d:ENTERO,

PARA (i=0:ENTERO DESDE j<tam2, HACER i++)

PARA (j=0:ENTERO DESDE j<tam2, HACER j++)

SI (primero[i]==segundo[j])

resultado[k]=primero[i]

k++

FIN SI

FIN PARA

FIN PARA

FIN SI

SI (k==0)ENTONCES

ESCRIBIR "No hay ningun elemento repetido"

FIN SI

PARA (a=0:ENTERO DESDE a<k HACER a++)

PARA (b=a+1:ENTERO DESDE b<k HACER b++)

SI (resultado[a] == resultado [b]) ENTONCES

d=b

MIENTRAS (d<k) ENTONCES

resultado[d] = resultado[d+1]

++d

FIN MIENTRAS

--k

--b

FIN SI

FIN PARA

FIN PARA

PARA (c=0:ENTERO DESDE c<k, ENTONCES c++)

ESCRIBIR "Elemento " c " arreglo resultado:" resultado[c]

FIN PARA

```

SI (tam1>tam2) ENTONCES
    i=0:ENTERO, j=0:ENTERO, k=0:ENTERO, d:ENTERO,
        PARA (i=0:ENTERO DESDE j<tam2, HACER i++)
            PARA (j=0:ENTERO DESDE j<tam1, HACER j++)
                SI (primero[i]==segundo[j])
                    resultado[k]=primero[i]
                    k++
                FIN SI
            FIN PARA
        FIN PARA
    FIN SI
    SI (k==0)ENTONCES
        ESCRIBIR "No hay ningun elemento repetido"
    FIN SI
    PARA (a=0:ENTERO DESDE a<k HACER a++)
        PARA (b=a+1:ENTERO DESDE b<k HACER b++)
            SI (resultado[a] == resultado [b]) ENTONCES
                d=b
                MIENTRAS (d<k) ENTONCES
                    resultado[d] = resultado[d+1]
                    ++d
                FIN MIENTRAS
                --k
                --b
            FIN SI
        FIN PARA
    FIN PARA
    PARA (c=0:ENTERO DESDE c<k, ENTONCES c++)
        ESCRIBIR "Se repiten los elemento " c " arreglo resultado:" resultado[c]
    FIN PARA
FIN FUC
FIN

```


CODIFICACIÓN

```
#include <stdio.h>
#include <string.h>
void encontrar(int primero[], int tam1, int segundo[], int tam2, int resultado[])
{
    if(tam1<tam2){
        int k=0;
        int i=0;
        int j=0;
        int d;
        for(i=0; i<tam1; i++){
            for(j=0; j<tam2; j++){
                if(primero[i]==segundo[j]){
                    resultado[k]=primero[i];
                    k++;
                }
            }
        }
        if(k==0){
            printf("\n%s", "No hay ningun elemento repetido");
            return;
        }
        for(int a = 0; a < k; a++){
            for(int b = a+1; b < k; b++){
                if(resultado[a] == resultado[b])
                {
                    d = b;
                    while(d < k)
                    {
                        resultado[d] = resultado[d+1];
                        ++d;
                    }
                    --k;
                    --b;
                }
            }
        }
    }
}
```

```

    }
        }
    }
    for(int c=0; c<k; c++){
        printf("\nElemento %d arreglo resultado ---> (%d)", c, resultado[c]);
    }
}
if(tam1>tam2){
    int k=0;
    int i=0;
    int j=0;
    int d;
    for(i=0; i<tam2; i++){
        for(j=0; j<tam1; j++){
            if(segundo[i]==primero[j]){
                resultado[k]=segundo[i];
                k++;
            }
        }
    }
}
if(k==0){
    printf("\n%s", "No hay ningun elemento repetido");
    return;
}
for(int a = 0; a < k; a++){
    for(int b = a+1; b < k; b++){
        if(resultado[a] == resultado[b])
        {
            d = b;
            while(d < k)
            {
                resultado[d] = resultado[d+1];
                ++d;
            }
            --k;
        }
    }
}

```

```

        --b;
    }
    }
}
for(int c=0; c<k; c++){
    printf("\nSe repiten:");
    printf("\nElemento %d arreglo resultado ---> %d", c, resultado[c]);
}
return;
}

```

PRUEBA DE ESCRITORIO

```

1  #include "../interseccion.c"
2
3  int main(){
4
5      printf("\nPrimer test:\n");
6      int primero[] = {5,9,2,4,6,0};
7      int segundo[] = {4,2,7,8,4,5,0};
8      int expResult[] = {};
9      encontrar(primero, 6, segundo, 7, expResult);
10
11     printf("\nSegundo test:\n");
12     int primero2[] = {22,89,4,10,33};
13     int segundo2[] = {1,2,9};
14     int expResult2[] = {};
15     encontrar(primero2, 5, segundo2, 3, expResult2);
16
17
18     return 0;
19 }

```

```

christiantrujillo@MacBook-Pro-de-Christian EjerciciosProyecto % gcc test/indexTest.c
christiantrujillo@MacBook-Pro-de-Christian EjerciciosProyecto % ./a.out test/indexTest.c

```

Primer test:

```

Elemento 0 arreglo resultado ----> (5)
Elemento 1 arreglo resultado ----> (2)
Elemento 2 arreglo resultado ----> (4)
Elemento 3 arreglo resultado ----> (0)

```

Segundo test:

No hay ningun elemento repetido

```

christiantrujillo@MacBook-Pro-de-Christian EjerciciosProyecto %

```