

INTERSECCIÓN

Objetivo: Se deberá crear un arreglo con los elementos que se repiten en dos arreglos dados. El resultado se guarda en el tercer arreglo dado.

Análisis del programa:

De acuerdo a los objetivos presentados del programa, para el desarrollo de nuestro código, analizaremos si a partir de dos arreglos dados podemos formar un tercero de acuerdo a los elementos que tengan incidencia en ambos de ellos.

Para nuestros datos de entrada tendremos en consideración tres arreglos de tipo entero en los que se encontrarán los elementos distribuidos, de manera que podamos leer y analizar sus componentes. De acuerdo a esto para nuestros datos de salida se nos presentaría un arreglo de tipo entero, resultante de los elementos que coincidieron en presentarse en ambos arreglos anteriores. De esta manera podríamos señalar fácilmente cuáles se repiten.

Pseudocódigo:

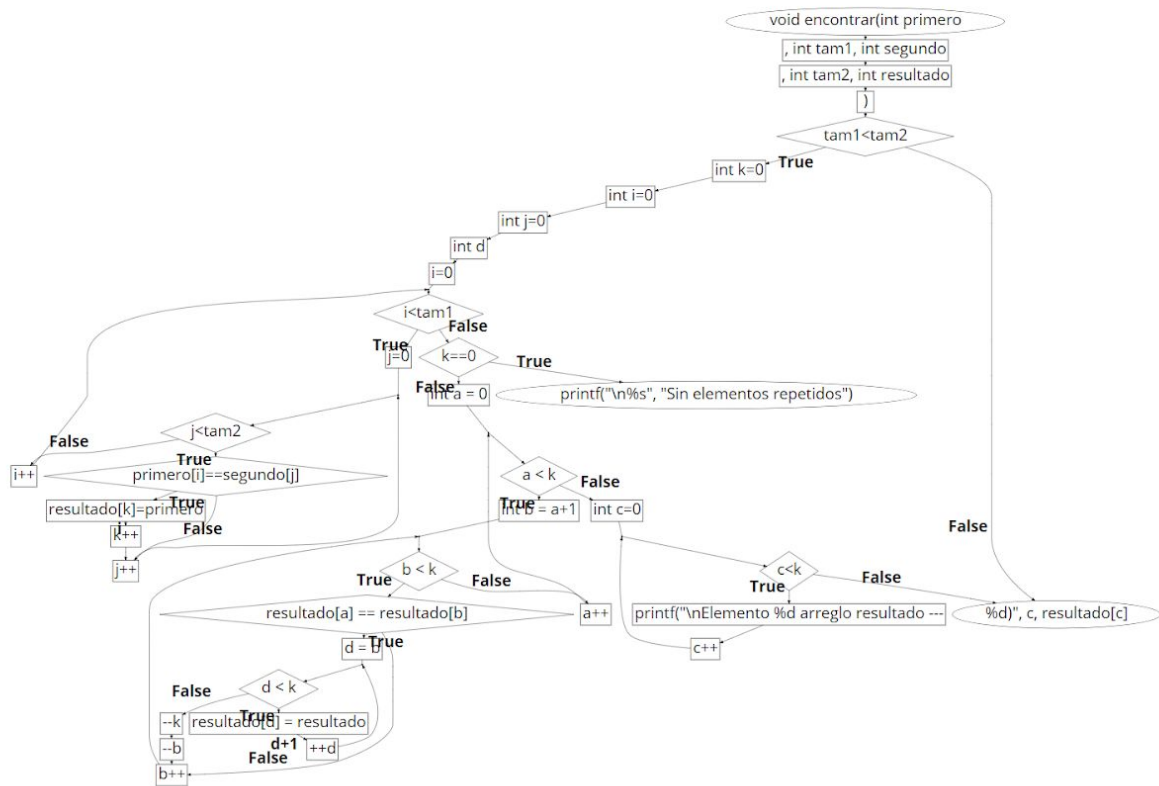
```
INICIO
    primero:ENTERO, tam1:ENTERO, segundo:ENTERO, tam2:ENTERO,
    resultados:ENTERO
    FUNC ENCONTRAR
        SI (tam1<tam2) ENTONCES
            i=0:ENTERO, j=0:ENTERO, k=0:ENTERO, d:ENTERO,
            PARA (i=0:ENTERO DESDE j<tam2, HACER i++)
                PARA (j=0:ENTERO DESDE j<tam2, 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 "Sin elementos repetidos"
    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
```

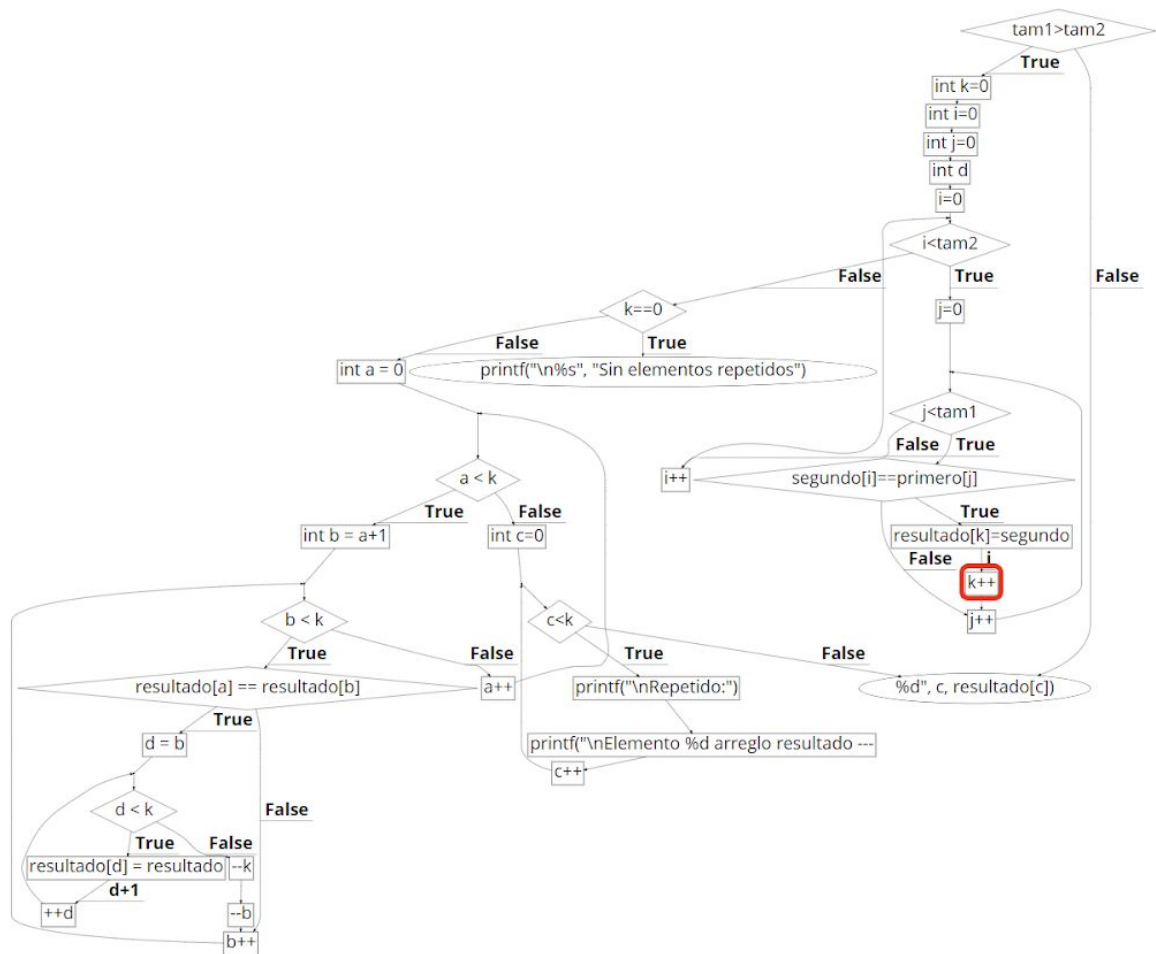
```

        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 "Sin elementos repetidos"
    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 "Elementos repetidos " c " arreglo resultado:"
    resultado[c] FIN PARA
    FIN FUC
    FIN

```

Diagrama de Flujo:





Código en C:

```

#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", "Sin elementos repetidos");
        }
    }
}

```

```

        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", "Sin elementos repetidos");
        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("\nRepetidos:");
        printf("\nElemento %d arreglo resultado ---> %d", c,
resultado[c]);
    }

    return;
}

```

```
// int main(){  
//     int primero[] = {1,2,2,3,4,5};  
//     int segundo[] = {2,7,8,9};  
//     int expResult[] = {};  
//     encontrar(primer, 6, segundo, 4, expResult);  
  
//     return 0;  
// }
```

Test:

```
Primer test:  
Elemento 0 arreglo resultado ---> (5)  
Elemento 1 arreglo resultado ---> (2)  
Elemento 2 arreglo resultado ---> (4)  
Elemento 3 arreglo resultado ---> (0)  
Segundo test:  
Sin elementos repetidos
```