

INTERSECCIÓN

ANÁLISIS

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

PSEUDOCÓDIGO DE INTERSECCIÓN

```
INICIO
p:ENTERO, t1:ENTERO, s:ENTERO, t2:ENTERO, r:ENTERO r:ENTERO
FUNC ENCONTRAR
SI (t1<t2) ENTONCES
    i=0:ENTERO, j=0:ENTERO, k=0:ENTERO, d:ENTERO,
    PARA (i=0:ENTERO DESDE j<t2, HACER i++)
        PARA (j=0:ENTERO DESDE j<t2, HACER j++)
            SI (p[i]==s[j])
                r[k]=p[i]
                k++
            FIN SI
        FIN PARA
    FIN PARA
FIN SI

SI (k==0)ENTONCES
    ESCRIBIR "No existe ningún elemento repetido"
FIN SI

PARA (a=0:ENTERO DESDE a<k HACER a++)
    PARA (b=a+1:ENTERO DESDE b<k HACER b++)
        SI (r[a] == r [b]) ENTONCES
            d=b
            MIENTRAS (d<k) ENTONCES
                r[d] = r[d+1]
                ++d
            FIN MIENTRAS
            --k
            --b
        FIN SI
    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 (t1>t2) ENTONCES
    i=0:ENTERO, j=0:ENTERO, k=0:ENTERO, d=ENTERO,
    PARA (i=0:ENTERO DESDE j<t2, HACER i++)
        PARA (j=0:ENTERO DESDE j<t1, HACER j++)
            SI (p[i]==segundo[j])
                resultado[k]=primero[i]
                k++
            FIN SI
        FIN PARA
    FIN PARA
FIN SI
```

```
SI (k==0)ENTONCES
    ESCRIBIR "No hay ningún elemento repetido"
FIN SI
```

```
PARA (a=0:ENTERO DESDE a<k HACER a++)
    PARA (b=a+1:ENTERO DESDE b<k HACER b++)
        SI (r[a] == r [b]) ENTONCES
            d=b
            MIENTRAS (d<k) ENTONCES
                r[d] = r[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
```

LENGUAJE ENC

```
#include<stdio.h>
```

```
int main {
```

```
    int p, t1, s, t2, r;
```

```
    if (t1<t2){
```

```
        int i,j,k,d;
```

```
        for (i=0;i<t2;i++){
```

```
            for (j=0;j<t2;j++){
```

```
                if (p[i]==s[j]){
```

```
                    r[k]=p[i];
```

```
                    k=k+1;
```

```
                }
```

```
            }
```

```
        }
```

```
    }
```

```
    if (k==0){
```

```
        printf("No existe ningún elemento repetido");
```

```
    }
```

```
    for (a=0;a<k;a++){
```

```
        for (b=0;b<k;b++){
```

```
            if(r[a]==r[b]){
```

```
                d=b;
```

```
                while(d<k){
```

```

        r[d]=r[d+1];

        d=d+1;

    }

    k=k-1;

    b=b-1;

}

}

}

for(c=0;c<k;c++){

    printf("%d"Elemento arreglo resultado "c);

}

if (t1<t2){

int i,j,k,d;

for (i=0;i<t2;i++){

    for (j=0;j<t2;j++){

        if (p[i]==s[j]){

            r[k]=p[i];

            k=k+1;

        }

    }

}

}

}

if (k==0){

```

```

        printf("No existe ningún elemento repetido");

    }

    for (a=0;a<k;a++){

        for (b=0;b<k;b++){

            if(r[a]==r[b]){

                d=b;

                while(d<k){

                    r[d]=r[d+1];

                    d=d+1;

                }

                k=k-1;

                b=b-1;

            }

        }

    }

    for(c=0;c<k;c++){

        printf("%d"Elemento arreglo resultado "c);

    }

    if (t1<t2){

        int i,j,k,d;

        for (i=0;i<t2;i++){

            for (j=0;j<t2;j++){

                if (p[i]==s[j]){

```

```

        r[k]==p[i];

        k=k+1;

    }

}

}

}

if (k==0){

    printf("No existe ningún elemento repetido");

}

for (a=0;a<k;a++){

    for (b=0;b<k;b++){

        if(r[a]==r[b]){

            d=b;

            while(d<k){

                r[d]=r[d+1];

                d=d+1;

            }

            k=k-1;

            b=b-1;

        }

    }

}

for(c=0;c<k;c++){

```

```
        printf("%d"Elemento arreglo resultado "c);  
    }  
}
```

PRUEBA DE ESCRITORIO

```
Prueba Uno  
Elemento 0 arreglo resultado --> 5  
Elemento 1 arreglo resultado --> 2  
Elemento 2 arreglo resultado --> 4  
Elemento 3 arreglo resultado --> 3  
Elemento 4 arreglo resultado -->0  
Prueba Dos  
No hay ningún elemento repetido  
SebastianZu@Pc-Proyecto-Interseccion-Pruebas%
```

TEST

```
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 }
```

```
#include "../interseccion.c"

int main (){

    printf("\nPrueba Uno:\n");

    int primero [] = {5,6,3,4,1,0,};

    int segundo [] = {4,3,8,2,4,5,0};

    int expResult [] = {};

    encontrar(primero, 5, segundo, 9, expResult);

    printf ("\nPrueba Dos:\n");

    int primero2 [] ={23,14,89,18,9,2};

    int segundo2 [] = {3,1,7};

    int expResult2 [] = {};

    encontrar(primero2,14,segundo2,1,expResult2);

    return 0;

}
```

DIAGRAMA DE FLUJO

