

```

/**Este Proyecto Estudia Posibles Preguntas para Entrevistas de Trabajo**/

#include <stdio.h>
#include <stdlib.h>

int main()
{
    /*swapWithout3Variables();
    callProceduresCorrespondingToArraySwappingComplexityON(); */
    callProceduresCorrespondingToArraySwappingComplexityO1();
    return 0;
}

/**Este Procedimiento Enseña como Intercambiar el Valor de 2 Variables, sin
Utilizar una 3era Variables**/

#include<stdio.h>
#include<stdlib.h>

void swapWithout3Variables()
{
    int a = 2, b = 3;
    printf("Valores antes del intercambio. \n\tA: %i\n\tB: %i\n", a, b);
    a = a + b;
    b = a - b;
    a = a - b;
    printf("Valores despues del intercambio. \n\tA: %i\n\tB: %i\n", a, b);
    printf("\n \n");
}

/**Este Procedimiento Intercambia la Informacion de 2 Vectores**/

/// La Complejidad de la Funcion es O(N)
// La complejidad es N porque para resolver el problema ejecutamos una
subrutina N veces

#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

#define N 5

void swap(int *iValue1, int *iValue2)
{

```

```

    int temp;
    temp = *iValue1;
    *iValue1 = *iValue2;
    *iValue2 = temp;
}

void swapONComplexity(int *iArray1, int *iArray2)
{
    int f;
    for (f = 0; f < N; f++)
        swap(&iArray1[f], &iArray2[f]);
}

void printArray(int *iArray)
{
    int f;
    printf("\t");
    for (f = 0; f < N; f++)
        printf("%i ", iArray[f]);
    printf("\n");
}

void callProceduresCorrespondingToArraySwappingComplexityON()
{
    int iArray1[N] = {1,2,3,4,5};
    int iArray2[N] = {10,9,8,7,6};
    printf("Impresion de Vectores antes del Intercambio. \n");
    printArray(iArray1);
    printArray(iArray2);
    swapONComplexity(iArray1, iArray2);
    printf("Impresion de Vectores Despues del Intercambio. \n");
    printArray(iArray1);
    printArray(iArray2);
    printf("\n");
}

/**Este Procedimiento Intercambia la Informacion de 2 Vectores**/

/// La Complejidad de la Funcion es O(1)
// La complejidad es 1 porque para resolver el problema ejecutamos una
subrutina 1 sola vez

#include<stdio.h>
#include<stdlib.h>

```

```
#include<stdbool.h>

#define iSIZE 5

int *generateArrayDynamically(int iSize)
{
    int f;
    int *iDynamicArray = malloc(iSize * sizeof(int));
    printf("Ingreso de la Informacion del Vector. \n");
    for (f = 0; f < iSize; f++)
    {
        printf("\tIngresa un elemento: ");
        scanf("%i", &iDynamicArray[f]);
    }
    printf("\n");
    return iDynamicArray;
}

void printArrayInformation(int *iArray)
{
    int f;
    printf("Impresion del Vector Completo: \n");
    for (f = 0; f < iSIZE; f++)
        printf("\tElemento #%i: %i\n", f + 1, iArray[f]);
    printf("\n");
}

void printArrayMemory(int *iArray)
{
    int f;
    printf("Direccion en Memoria del Vector Completo: %i\n", &iArray);
    for (f = 0; f < iSIZE; f++)
        printf("\tDireccion en memoria del elemento #%i: %i\n", f + 1,
&iArray[f]);
    printf("\n");
}

void swap1NComplexity(int **iArray1, int **iArray2)
{
    int *temp;
    temp = *iArray1;
    *iArray1 = *iArray2;
    *iArray2 = temp;
}
```

```
void callProceduresCorrespondingToArraySwappingComplexity01()
{
    int *iArray1, *iArray2;

    iArray1 = generateArrayDynamically(iSIZE);
    iArray2 = generateArrayDynamically(iSIZE);

    printf("Impresion de Vectores antes del Intercambio de Informacion.
\n");
    printArrayInformation(iArray1);
    printArrayInformation(iArray2);

    swap1NComplexity(&iArray1, &iArray2);

    printf("Impresion de Vectores despues del Intercambio de Informacion.
\n");
    printArrayInformation(iArray1);
    printArrayInformation(iArray2);

    printf("\n \n");
}
```