/\*\*Este Proyecto Contiene Ejercicios Practicos de Vectores\*\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

    duplicateArray();

    /\*palindromeOrNot();

    largestNeighbours();

    arraySortedOrNot();

    uniqueElements();

    repeatedElements();

    rotateElementsLeft();

    rotateElementsRight();

    nearestToZero();

    printSumOfVector(); \*/

    printFreshVector();

    return 0;

}

/\*\*Este Procedimiento Copia la Informacion de un Vector y la Asigna a otro Vector\*\*/

void duplicateArray()

{

    int iArray[3] = {26,03,2004}, iCopyArray[3], f;

    printf("Vector Original\tVector Duplicado.\n");

    for (f = 0; f < 3; f++)

    {

        iCopyArray[f] = iArray[f];

        printf("- %i\t\t- %i\n", iArray[f], iCopyArray[f]);

    }

    printf("\n \n");

}

/\*\*Este Procedimiento Valida si un Vector Consituye un Pal�ndromo o No\*\*/

#include<stdio.h>

#define iSIZE 8

int palindromeOrNot()

{

    int iArray[iSIZE], f;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    for (f = iSIZE - 1; f >= 0; f--)

    {

        if (iArray[f] != iArray[ ((iSIZE - 1) - f) ] )

        {

            printf("El vector ingresado no es un palindromo.\n\n");

            return 0;

        }

    }

    printf("El vector ingresado es un palindromo. ");

    printf("\n \n");

    return 0;

}

/\*\*Este Procedimiento Lee del Usuario un Vector y Procede a Imprimir la Suma m�s Grande Formada por Elementos Adyacentes\*\*/

#include<stdio.h>

#define iSIZE 5

void largestNeighbours()

{

    int iArray[iSIZE], f, iSum;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

        if (f != 0)

        {

            if (f == 1)

                iSum = iArray[0] + iArray[1];

            else if (iArray[f - 1] + iArray[f] > iSum)

                iSum = iArray[f - 1] + iArray[f];

        }

    }

    printf("La suma mas grande formada por la adicion de 2 elementos del vector es: %i", iSum);

    printf("\n\n");

}

/\*\*Este Procedimiento Verifica si Un Vector es Ascendente, Ordenado o no Tiene Orden\*\*/

#include<stdio.h>

#define iSIZE 6

// iArray = {1,2,5,8,10} --> Ascendente

// iArray = {1,2,2,3,11} --> Ordenado

// iArray = {3,2,5,6,4}  --> Sin Orden

void arraySortedOrNot()

{

    int iArray[iSIZE], f, lSorted = 1;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

        if (f != 0)

        {

            if (iArray[f - 1] < iArray[f] && lSorted != -1)

                lSorted++;

            else if (iArray[f - 1] == iArray[f] && lSorted != -1)

                lSorted = 1;

            else if (iArray[f - 1] > iArray[f])

                lSorted = -1;

        }

    }

    if (lSorted == iSIZE)

        printf("El vector tiene una alineacion ascendente. ");

    else if (lSorted > -1 && lSorted < iSIZE)

        printf("El vector esta ordenado. ");

    else

        printf("El vector no tiene orden. ");

    printf("\n \n");

}

/\*\*Este Procedimiento se Encarga de Imprimir y Contar los Elementos Unicos de un Vector\*\*/

#include<stdio.h>

#define iSIZE 10

void uniqueElements()

{

    int iArray[iSIZE], iCounter = 0, f, k;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un numero entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("Impresion de los Valores Unicos:\n");

    for (f = 0; f < iSIZE; f++)

    {

        for (k = 0; k < iSIZE; k++)

        {

            if (k == f)

                continue;

            if (iArray[f] == iArray[k])

                break;

        }

        if (k == iSIZE)

        {

            printf("\tNumero Unico: %i\n", iArray[f]);

            iCounter++;

        }

    }

    if (iCounter == 0)

        printf("No hay valores unicos en el vector. ");

    else

        printf("Existen %i numero(s) unicos en el vector ingresado. ", iCounter);

    printf("\n \n");

}

/\*\*Este Procedimiento se Encarga de Imprimir y Contar los Numeros Repitidos de un Vector\*\*/

#define iSIZE 10

#include<stdio.h>

void repeatedElements()

{

    int iArray[iSIZE], f, k, iCounter = 0, lRepeatedNotIntoAccount = 0;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("Impresion de Valores Repetidos: \n");

    for (f = 0; f < iSIZE; f++)

    {

        for (k = 0; k < iSIZE; k++)

        {

            if (k == f)

                continue;

            if (iArray[f] == iArray[k] && k > f)

                break;

            else if (iArray[f] == iArray[k] &&  k < f)

            {

                lRepeatedNotIntoAccount = 1;

                break;

            }

        }

        if (k != iSIZE && lRepeatedNotIntoAccount == 0)

        {

            printf("\t- %i\n", iArray[f]);

            iCounter++;

        }

        else

            lRepeatedNotIntoAccount = 0;

    }

    if (iCounter == 0)

        printf("No hay ningun elemento repetido en el vector. ");

    else

        printf("Hay un total de %i elemento(s) repetido(s) en el vector. ", iCounter);

    printf("\n \n");

}

/\*\*Este Procedimiento Rota hacia la Izquierda los Elementos de un Vector \*\*/

#define iSIZE 5

#include<stdio.h>

void rotateElementsLeft()

{

    int iArray[iSIZE], f, k, iAuxiliar, iN;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("Impresion del Vector Ingresado: \n\t");

    for (f = 0; f < iSIZE; f++)

        printf("%i ", iArray[f]);

    do{

        printf("\nIngresa el numero de posiciones que deseas rotar el vector hacia la izquierda (mayor a cero): ");

        scanf("%i", &iN);

        fflush(stdin);

        if (iN <= 0)

            printf("ERROR: El numero de posiciones debe ser mayor a cero. ");

    } while (iN <= 0);

    for (k = 1; k <= iN; k++)

    {

        iAuxiliar = iArray[0];

        for (f = 1; f < iSIZE; f++)

        {

            iArray[f - 1] = iArray[f];

        }

        iArray[iSIZE - 1] = iAuxiliar;

    }

    printf("\nImpresion del Vector con una Rotacion de Elementos hacia la Izquierda: \n\t");

    for (f = 0; f < iSIZE; f++)

        printf("%i ", iArray[f]);

    printf("\n\n");

}

/\*\*Este Procedimiento se Encarga de Rotar los Elementos de un Vector hacia la Derecha\*\*/

#define iSIZE 5

#include<stdio.h>

void rotateElementsRight()

{

    int iArray[iSIZE], f, k, iAuxiliar, iN;

    printf("Ingreso de %i Elementos. \n", iSIZE);

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa el valor #%i: ", f + 1);

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("Impresion del Vector con los Elementos Ingresados: \n \t");

    for (f = 0; f < iSIZE; f++)

        printf("%i ", iArray[f]);

    do{

        printf("\nIngresa el numero de posiciones que deseas rotar el vector hacia la derecha (mayor a cero): ");

        scanf("%i", &iN);

        fflush(stdin);

        if (iN <= 0)

            printf("ERROR: El numero de posiciones debe ser mayor a cero. ");

    } while (iN <= 0);

    printf("Impresion del Vector con los Elementos Rotados hacia la Derecha %i Posicion(es): \n\t", iN);

    for (k = 1; k <= iN; k++)

    {

        iAuxiliar = iArray[iSIZE - 1];

        for (f = iSIZE - 2; f >= 0; f--)

            iArray[f + 1] = iArray[f];

        iArray[0] = iAuxiliar;

    }

    for (f = 0; f < iSIZE; f++)

        printf("%i ", iArray[f]);

    printf("\n \n");

}

/\*\*Este Procedimiento se Encarga de Encontrar, dentro de los Elementos de un Vector, la Suma de 2 Numeros m�s Cercano al Cero\*\*/

#define iSIZE 6

#include<stdio.h>

int absoluteValue(int iValue)

{

    if (iValue >= 0)

        return iValue;

    else

        return iValue \* -1;

}

void nearestToZero()

{

    int iArray[iSIZE], f, k, iNearest;

    printf("Ingreso de %i Elementos Enteros. \n", iSIZE);

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa el valor entero #%i: ", f + 1);

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("La suma de 2 numeros mas cercana a cero es: ");

    iNearest = iArray[0] + iArray[1];

    for (f = 0; f < iSIZE - 1; f++)

    {

        for (k = f + 1; k < iSIZE; k++)

        {

            if ( absoluteValue(iArray[f] + iArray[k]) < absoluteValue(iNearest))

                iNearest = iArray[f] + iArray[k];

        }

    }

    printf("%i. ", iNearest);

    printf("\n \n");

}

/\*\*Este Procedimiento Calcula la Suma de Todos los Elementos de un Vector\*\*/

#define iSIZE 10

#include<stdio.h>

int calculateSumOfVector(int iArray[iSIZE])

{

    int f, iSum = 0;

    for (f = 0; f < iSIZE; f++)

        iSum = iSum + iArray[f];

    return iSum;

}

void printSumOfVector()

{

    int iArray[iSIZE], f;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa el valor entero #%i: ", f + 1);

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("La suma de los elementos del vector es: %i", calculateSumOfVector(iArray));

    ///NOTA: La sintaxis, "iArray", denota la direccion de memoria del primer elemento del vector

    /\*printf("La direccion del primer elemento del vector: %i", iArray);

    printf("\nImpresion de las Direcciones de Memoria de los Elementos del Vector.");

    for (f = 0; f < iSIZE; f++)

        printf("\n#%i: %lu", f + 1, &iArray[f]);\*/

    printf("\n \n");

}

/\*\*Este Procedimiento Lee un Vector y Procede a Reiniciar Todos sus Elementos a Cero\*\*/

#define iSIZE 5

#include<stdio.h>

void inputArray(int \*pElementOfArray)

{

    int f;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa el valor entero #%i: ", f + 1);

        scanf("%i", &pElementOfArray[f]);

        fflush(stdin);

    }

}

void outputArray(int \*pElementOfArray)

{

    int f;

    for (f = 0; f < iSIZE; f++)

        printf("\n%i", pElementOfArray[f]);

    printf("\n");

}

void freshVector(int \*pElementOfArray)

{

    int f;

    for (f = 0; f < iSIZE; f++)

        pElementOfArray[f] = 0;

}

void printFreshVector()

{

    int iArray[iSIZE];

    inputArray(iArray);

    printf("Impresion de Vector Ingresado.");

    outputArray(iArray);

    freshVector(iArray);

    printf("Impresion de Vector Reseteado.");

    outputArray(iArray);

    printf("\n \n");

}