/\*\*Este Proyecto Explora el Tema de Vectores de 1 Dimension\*\*/

#include <stdio.h>

#include <stdlib.h>

int main()

{

    /\*initializeArrays();

    worstGrade();

    maxValueOfArray();

    weatherStation();

    goodNeighbour(); \*/

    displayReverse();

    return 0;

}

/\*\*Este Procedimiento Muestra como Inicializar Vectores\*\*/

void initializeArrays()

{

    int f;

    //Inicializar un Vector con Todos los Elementos en 0

    int iArrayA[5] = {0};

    for (f = 0; f <= 4; f++)

        printf("Vector A[%i]: %i\n", f, iArrayA[f]);

    printf("\n");

    //Inicializar los Primeros 2 Elementos de un Vector en 0, Mientras que los Dem�s Elementos Quedan Inicializados en 0

    int iArrayB[5] = {1,2};

    for (f = 0; f <= 4; f++)

        printf("Vector B[%i]: %i\n", f, iArrayB[f]);

    printf("\n\n");

}

/\*\*Este Procedimiento se Encarga de Accesar y Modificar los Elementos de un Vector\*\*/

void worstGrade()

{

    float fGrades[6], fWorstGrade;

    int f;

    for (f = 0; f < 6; f++)

    {

        do{

            printf("Ingresa la califiacion[%i] (1-100): ", f + 1);

            scanf("%f", &fGrades[f]);

            if ( !(fGrades[f] >= 1 && fGrades[f] <= 100) )

                printf("ERROR: La calificacion ingresada debe ser entre 1 y 100. \n");

            else if (f > 0)

            {

                if (fGrades[f] < fWorstGrade)

                    fWorstGrade = fGrades[f];

            }

            else if (f == 0)

                fWorstGrade = fGrades[0];

        }while ( !(fGrades[f] >= 1 && fGrades[f] <= 100) );

    }

    printf("\nLa peor calificacion es: \n");

    for (f = 0; f < 6; f++)

    {

        printf("- %0.2f", fGrades[f]);

        if (fWorstGrade == fGrades[f])

            printf(" <---");

        printf("\n");

    }

    printf("\n \n");

}

/\*\*Este Procedimiento Se Encargar de Encontrar el Maximo Valor dentro de un Vector\*\*/

void maxValueOfArray()

{

    int iValues[10], f, iMaxValue;

    for (f = 0; f < 10; f++)

    {

        do{

            printf("Ingresa un valor entero entre 0 y 9: ");

            scanf("%i", &iValues[f]);

            if (iValues[f] < 0 || iValues[f] > 9)

                printf("ERROR: El valor ingresado no se encuentra entre 0 y 9. \n");

        } while (iValues[f] < 0 || iValues[f] > 9);

        if (f > 0)

        {

            if (iMaxValue < iValues[f])

                iMaxValue = iValues[f];

        }

        else if (f == 0)

            iMaxValue = iValues[0];

    }

    printf("\nEl mayor valor es: \n");

    for (f = 0; f < 10; f++)

    {

        printf("- %i", iValues[f]);

        if (iMaxValue == iValues[f])

            printf(" <--- Indice del Valor: %i", f);

        printf("\n");

    }

    printf("\n \n");

}

/\*\*Este Procedimiento Lee del Usuario 7 Temperaturas y Calcula Tanto el Promedio, As� Como m�s Informaci�n\*\*/

#include <stdio.h>

#define iSIZE 7

void weatherStation()

{

    float fTemp[iSIZE], fAverage, fMin, fMax;

    int f;

    for (f = 0; f < iSIZE; f++)

    {

        do{

            printf("Ingresa la temperatura Celsius del dia %i: ", f + 1);

            scanf("%f", &fTemp[f]);

            fflush(stdin);

            if (fTemp[f] < -273.15)

                printf("ERROR: La temperatura celsius ingresada no existe. \n");

            else

            {

                fAverage = fAverage + fTemp[f];

                if (f == 0)

                {

                    fMin = fTemp[f];

                    fMax = fTemp[f];

                }

                else if (f > 0)

                {

                    if (fMin > fTemp[f])

                        fMin = fTemp[f];

                    if (fMax < fTemp[f])

                        fMax = fTemp[f];

                }

            }

        }while (fTemp[f] < -273.15);

    }

    printf("\nInformaci�n Calculada: \n");

    for (f = 0; f < iSIZE; f++)

    {

        printf("Dia %i: %0.2f Celsius", f + 1, fTemp[f]);

        if (fTemp[f] == fMin)

            printf(" <-- Temperatura minima");

        if (fTemp[f] == fMax)

            printf(" <-- Temperatura maxima");

        printf("\n");

    }

    fAverage = fAverage / iSIZE;

    printf("El promedio de las temperaturas es %0.2f. ", fAverage);

    printf("\n \n");

}

/\*\*Este Procedimiento Determina si Hay Buenos Vecinos(Elementos que equivalen a la Multiplicaci�n de sus Elementos Adyacentes a la Derecha y a la Izquierda) dentro de un Vector\*\*/

#include<stdio.h>

#define iSIZE1 5

int goodNeighbour()

{

    int iArray[iSIZE1], f;

    for (f = 0; f < iSIZE1; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    for (f = 1; f < iSIZE1 - 1; f++)

    {

        if (iArray[f - 1] \* iArray[f + 1] == iArray[f])

        {

            printf("La lista de elementos ingresados contiene un buen vecino. \n \n");

            return 0;

        }

    }

    printf("La lista de elementos ingresados no contiene un buen vecino. ");

    printf("\n \n");

    return 0;

}

/\*\*Este Procedimiento Lee del Usuario los Elementos de un Vector y Procede a Desplegarlos en un Orden de Reversa\*\*/

#include<stdio.h>

#define iSIZE 10

void displayReverse()

{

    int iArray[iSIZE], f;

    for (f = 0; f < iSIZE; f++)

    {

        printf("Ingresa un valor entero: ");

        scanf("%i", &iArray[f]);

        fflush(stdin);

    }

    printf("\nImpresion de los valores ingresados. \n");

    for(f = 0; f < iSIZE; f++)

        printf("\t %i\n", iArray[f]);

    printf("\nImpresion de los valores ingresados en orden invertido. \n");

    for(f = iSIZE - 1; f >= 0; f--)

        printf("\t %i\n", iArray[f]);

    printf("\n \n");

}