|  |  |
| --- | --- |
|  | **ESCUELA POLITÉCNICA NACIONAL - ESFOT**  **PROGRAMACIÓN. TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE.**  **PRUEBA 2B-1**  **Quito, 16 de agosto de 2022.**  **NOMBRE:** |

**Instrucciones:**

Duración: Esta prueba es individual y tiene una duración de una hora y 45 minutos.

Envío: La prueba se realizará en este archivo de Microsoft Word, los problemas pueden ser realizado en Visual Studio o Repl.it, pero se debe copiar el código en este documento, guardarlo como Word(.docx) y subirlo al aula virtual antes de que se cumplan el tiempo desde del inicio de la prueba. Los dos ejercicios tienen el mismo puntaje.

**Problemas:**

1. Declarar dos matrices multidimensionales de 3x3 y realizar la suma y la resta de matrices, el resultado se almacenará en otras dos matrices multidimensionales respectivamente. Imprimir las matrices iniciales y las resultantes.

Calendario

Descripción generada automáticamente

#include<iostream>

using namespace std;

int main(){

int matrizA[3][3]= {2, 0, 0, 3, 0, 0, 5, 1, 1};

int matrizB[3][3]={1, 0, 1, 1, 2, 1, 1, 1, 0};

cout << "LA MATRIZ 'A' ES "<< endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << matrizA[i][j] <<" ";

}

cout << endl;

}

cout << "LA MATRIZ 'B' ES "<< endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << matrizB[i][j] <<" ";

}

cout << endl;

}

cout << "LA SUMA DE LA MATRIZ A+B ES "<< endl;

int matrizSum[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j <3 ; j++)

{

matrizSum [i][j] = matrizA[i][j] + matrizB[i][j];

cout << matrizSum[i][j] << " ";

}

cout << endl;

}

cout << "LA RESTA DE LA MATRIZ ABB ES "<< endl;

int matrizRes[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j <3 ; j++)

{

matrizRes [i][j] = matrizA[i][j] - matrizB[i][j];

cout << matrizRes[i][j] << " ";

}

cout << endl;

}

system("pause");

return 0;

}

**2.** Se pide construir un programa que procese una lista de canciones favoritas como una lista de longitud variable de canciones (máximo 10). Para cada canción se mantiene la siguiente información:

* Número de canción (entero)
* Título (varias palabras)
* Artista (varias palabras)
* Genero (una sola palabra)
* Año (int)

El programa comenzará leyendo la información de las canciones de un archivo música.txt. El archivo contiene cinco líneas por canción. Termina con un ‐1 como el número de canción. Si hay más del máximo de canciones, se ignorarán las adicionales.

El programa en su menú permitirá listar todas las canciones, buscar canciones por artista, añadir al final de la lista nuevas canciones si hay menos del máximo y salir. Estas funcionalidades se realizarán mediante funciones, cuando se seleccione salir se guardarán las canciones añadidas en el archivo, para que la próxima vez que se ejecute el programa se carguen las canciones añadidas.

**#include <iostream>**

**#include <fstream>**

**#include<iomanip>**

**using namespace std;**

**const int MAX = 10;**

**typedef struct**

**{**

**int num\_canciones;**

**string titulo;**

**string artista;**

**string genero;**

**int anio;**

**} tcanciones[MAX];**

**int menuCanciones();**

**void enlistarCanciones();**

**int op, seguir;**

**int main()**

**{**

**fstream archivo;**

**archivo.open("musica.txt");**

**if (archivo.is\_open())**

**{**

**cout << setw(20) <<"El archivo se abrio" << endl;**

**}**

**else**

**{**

**cout << setw(20) << "El archivo se NO SE PUDO ABRIR" << endl;**

**}**

**do**

**{**

**op = menuCanciones();**

**switch (op)**

**{**

**case 1:**

**cout << "1. Lista canciones" << endl;**

**enlistarCanciones();**

**break;**

**case 2:**

**cout << "2. Buscar canciones por artista" << endl;**

**break;**

**case 3:**

**cout << "3. Aniadir canciones" << endl;**

**break;**

**case 4:**

**cout << "Saliendo...." << endl;**

**break;**

**default:**

**cout << "La opcion no esta en el rango";**

**break;**

**}**

**cout << "Desea seguir en el programa? 1.SI ";**

**cin >> seguir;**

**} while (seguir != 0);**

**system("pause");**

**return 0;**

**}**

**int menuCanciones()**

**{**

**int opcion;**

**cout << "======== Menu canciones ====== " << endl;**

**cout << "1. Lista canciones" << endl;**

**cout << "2. Buscar canciones por artista" << endl;**

**cout << "3. Aniadir canciones" << endl;**

**cout << "4. Salir" << endl;**

**cout << "Elija una opcion: ";**

**cin >> opcion;**

**return opcion;**

**}**

**void enlistarCanciones(){**

**bool ok = true;**

**char aux;**

**tcanciones canciones;**

**ofstream archivo;**

**archivo.open("musica.txt");**

**if (!archivo.is\_open())**

**{**

**ok = false;**

**}else**

**{**

**cout<<"Numero de canciones";**

**archivo << canciones->num\_canciones;**

**while (canciones->num\_canciones != -1)**

**{**

**archivo << canciones->num\_canciones;**

**}**

**}**

**archivo.close();**

**}**

**void mostrarCancion(int pos, tcanciones cancion) {**

**cout << setw(3) << cancion->num\_canciones**

**<< setw(20) << cancion->anio**

**<< setw(20) << cancion->artista**

**<< setw(20) << cancion->genero<< endl;**

**}**

Contenido del archivo música.txt:

1

Natural

movimiento original

rap

2016

2

El vals del proletariado

Skap

Skap

2003

3

Adios

aventura

bachata

2002

-1