

Informe: arreglos “migraciones.txt”

Profesor: Jonathan Frez

Ayudante: Michiru Nakamura

Nombre: Gonzalo Salazar

Sección:5

Introducción

En este informe se describirán los procedimientos realizados para completar el programa a realizar. Para lograrlo, se necesitarán conocimiento de arreglos, manejo de datos y manipulación de archivos externos. Los arreglos facilitan el almacenamiento de datos ordenadamente en la memoria RAM, esto es necesario para poder contener una gran cantidad de información y facilitar la lectura de la misma; Existirá un gran uso de manejo de datos para poder operar con la memoria y modificar algunas variables; Se utilizará manejo de archivos externos, los cuales permiten la escritura(exportación) y la lectura(importación) de grandes cantidades de datos, sin la necesidad de escribirlo en consola mediante la función *cin* o *cout* respectivamente, permitiendo el posterior uso de estos datos en otros programas.

Descripción de Solución

Para resolver los problemas de la actividad fue necesario utilizar la librería *fstream* para el manejo de archivos externos y la librería *sstream* para permitir la concatenación de datos obtenidos en los arreglos del archivo “migraciones.txt”.

Las funciones clave en la solución fueron el ciclo *while* realizado para que el programa lea el archivo hasta que termine, ya que si se utilizaba un ciclo *for* el programa no era capaz de leer la totalidad de datos, y la función *istringstream* la cual nos permitía manejar los valores de determinadas columnas del archivo, que se encuentran en variable *string* y manipularlas en variables numéricas, *int* en este caso, para realizar las sumatorias correspondientes en cada uno de los problemas a resolver.

Desarrollo

Definimos las librerías a utilizar, además de agregar el uso de un parámetro específico (std::stringstream) para que el programa compile sin errores.

#include <iostream>

#include <fstream>

#include <sstream>

using std::stringstream;

using namespace std;

Declaramos el inicio de la función principal.

int main(){

int muertes=-1,desaparecidas=0,md=0,ms=0,casosauto=0,casosdrow=0,casosshot=0;

double tablamuertes,tabladesaparecidos,tablalat,tablalon,latdrow,londrow,latcar,loncar;

string p1,p2,p3,p4,p5,p6;

Declaramos las variables de valores enteros, que utilizaremos para las sumatorias en los problemas más adelante.

Declaramos las variables de valores decimales, en *double* para que pueda almacenar valores numéricos más grandes y así mostrar bien resultados.

Declaramos las variables de tipo *string* que almacenarán cada una de las columnas del archivo.

Definimos la función llamada *archivo* que ejecutará “migraciones.txt”.

ifstream archivo("migraciones.txt");

Utilizamos el condicional para que el programa detecte si el archivo “migraciones.txt” existe.

if (archivo.fail()){

cout<<"\nNo se encuentra el archivo migraciones.txt\nPor favor descarguelo desde el grupo de Facebook y ubiquelo junto al programa.";

return 0;

}

Si el archivo no se encuentra, se mostrará en pantalla dónde se puede descargar el archivo.

Si no se encuentra el archivo, el programa finalizará.

Se señala que mientras el archivo no termine, continúe ejecutando los comandos que le siguen.

while(!archivo.eof()){

archivo>>p1>>p2>>p3>>p4>>p5>>p6;

istringstream(p3)>>tablamuertes;

istringstream(p4)>>tabladesaparecidos;

istringstream(p5)>>tablalat;

istringstream(p6)>>tablalon;

Se indica el orden que siguen los arreglos dentro del archivo.

Se indica que los valores *string* de la columna relacionada a muertes, se guarden como valores enteros.

Se indica que los valores *string* de la columna relacionada a desaparecidos, se guarden como valores enteros.

Se indica que los valores *string* de la columna relacionada a latitud, se guarden como valores decimales.

Se indica que los valores *string* de la columna relacionada a longitud, se guarden como valores decimales.

Se almacena la sumatoria de todos los valores de la columna relacionada a muertes. (SOLUCIÓN 1)

muertes+=tablamuertes;

desaparecidas+=tabladesaparecidos;

Se almacena la sumatoria de todos los valores de la columna relacionada a desaparecidos. (SOLUCIÓN 2)

if(p1=="drowned"){

casosdrow++;

md+=tablamuertes;

latdrow+=tablalat;

londrow+=tablalon;

dd+=tabladesaparecidos

}

Se define un condicional que encuentre la palabra “drowned” en la columna ligada a causas.

Se realiza una sumatoria de las veces que se encuentra la palabra, para luego usarla como contador.

Se almacena la sumatoria de los valores de la columna muertes relacionados a “drowned”. (SOLUCIÓN 3)

Se almacena la sumatoria de los valores de la columna latitud relacionados a “drowned”.

Se almacena la sumatoria de los valores de la columna longitud relacionados a “drowned”.

Se almacena la sumatoria de los valores de la columna desaparecidos relacionados a “drowned”. (SOLUCION 5)

Se define un condicional que encuentre la palabra “shot\_by\_the\_police” en la columna ligada a causas.

if(p1=="shot\_by\_the\_police"){

casosshot++;

ms+=tablamuertes;

}

Se realiza una sumatoria de las veces que se encuentra la palabra, para luego usarla como contador.

Se almacena la sumatoria de los valores de la columna muertes relacionados a “shot\_by\_the\_police”. (SOLUCIÓN 4)

Se define un condicional que encuentre la palabra “car\_accident” en la columna ligada a causas .

if(p1=="car\_accident"){

casosauto++;

latcar+=tablalat;

loncar+=tablalon;

}

if(tabladesaparecidos>2){

causa=p1;

}

}

Se realiza una sumatoria de las veces que se encuentra la palabra, para luego usarla como contador.

Se almacena la sumatoria de los valores de la columna latitud relacionados a “car\_accident”.

Se almacena la sumatoria de los valores de la columna longitud relacionados a “car\_accident”.

Se define un condicional que analice los valores enteros de la columna de desaparecidos.

Almacena la causa ligada al condicional definido.

\*\*Se cierra el ciclo *while*\*\*

Se muestran en pantalla las sumatorias realizadas, mostrando las soluciones en orden (1,2,3,4,5).

cout<<muertes<<endl<<desaparecidas<<endl<<md<<endl<<ms<<endl<<causa<<endl;

cout<<latdrow/casosdrow<<" "<<londrow/casosdrow<<endl;

cout<<latcar/casosauto<<" "<<loncar/casosauto;

Se realiza la operación para calcular el promedio “drowned”. (SOLUCIÓN 6)

Se realiza la operación para calcular el promedio “car\_accident”. (SOLUCIÓN 7)

archivo.close();

return 0;

}

Se indica a la función que contiene el “migraciones.txt” lo cierre. Se indica que el programa finaliza.

\*\*Termina la función *main*.\*\*

Resultados y conclusiones

Se puede concluir que el manejo de arreglos es bastante útil para el almacenamiento de muchos datos y que un solo error puede modificar todo lo que este incluye; además poder manejar archivos externos resulta muy cómodo para realizar análisis de datos.

Respecto a los resultados entregados por el programa, es posible decir que las muertes de migrantes que intentaba acceder a Europa desde el año 2000 a 2016 alcanza la gran cifra de 6918, siendo ahogo la causa con más muertes provocadas, con un total de 3605, cuya latitud y longitud promedio es 34.8386 y 8.67206 respectivamente; de un total de 127 casos de accidentes de tránsito se pudo estimar la latitud y longitud promedio las cuales son 41.9565 y 14.9543 respectivamente; hubo un total de 6265 desaparecidos, siendo ahogo la causa con más desapariciones, con un total de 5293.