Pablo de la Mora Vega

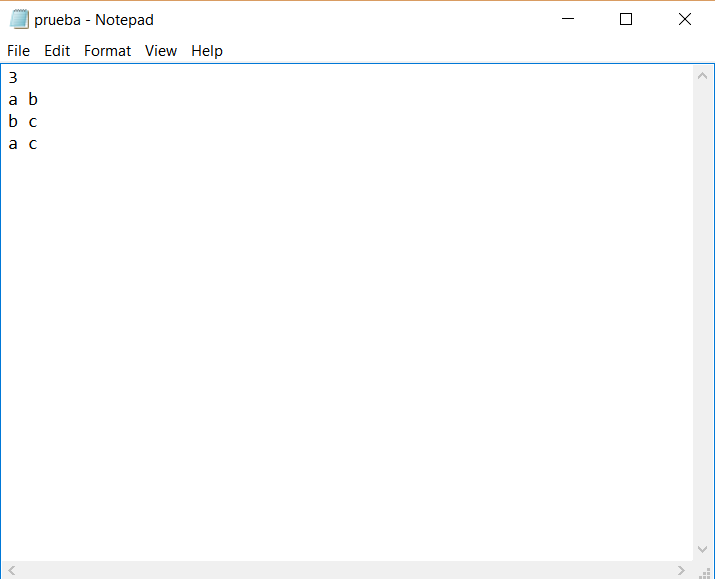
A01020365

Reporte

Proyecto 1

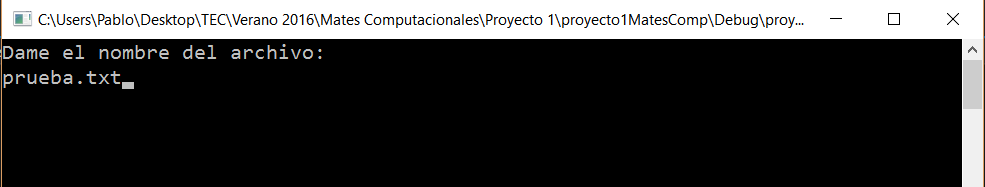
Manual de Usuario:

Este es un programa programado en C++, el cual se necesita tener un archivo de texto en la misma carpeta donde se ubica el código. El archivo tiene que ser .txt y consta de 2 partes. La primera línea tiene que ser solo un número que indique el número de relaciones binarias que contiene el archivo. Una relación binaria tan solo consta de 2 números separados por un espacio en esta ocasión. Entonces el archivo se tiene que ver con este formato:



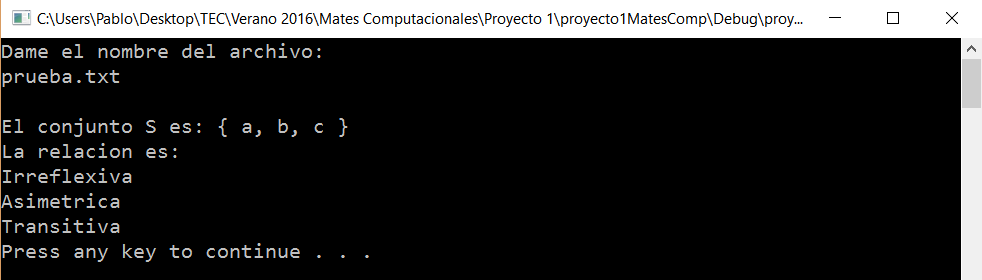
Puede contener letras o números (a excepción de la primera línea que necesita forzosamente el número de relaciones que contiene el archivo.

Ya que se tiene este archivo en la carpeta se corre el código y te va a pedir el nombre del archivo de texto descrito anteriormente. Se necesita escribir el nombre del archivo agregándole .txt al final. En la imagen anterior el archivo se llama prueba por lo que se tendrá que escribir prueba.txt como se muestra en esta imagen:



Después de haber realizado este paso el programa te dirá los elementos del conjunto S y te dirá si la relación es reflexiva, irreflexiva, simétrica, asimétrica, transitiva o que ninguna cumple en dicho caso.

Este es el output que da el programa al correrlo con el archivo de la primera imagen:



En esta ocasión se muestra que tienen una relación irreflexiva, asimétrica y transitiva.

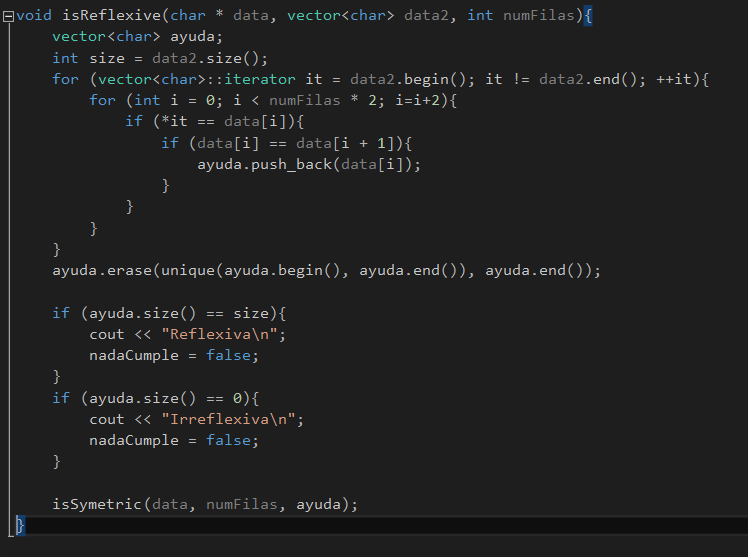
Descripción Técnica del Código:

Este código está programado en C++. Para empezar se utiliza el ifstream para poder leer el archivo y almacenar todos los datos necesarios en una variable dinámica.

Para poder desplegar los elementos del conjunto tan solo se usaron las librerías de vector y de algorithm.

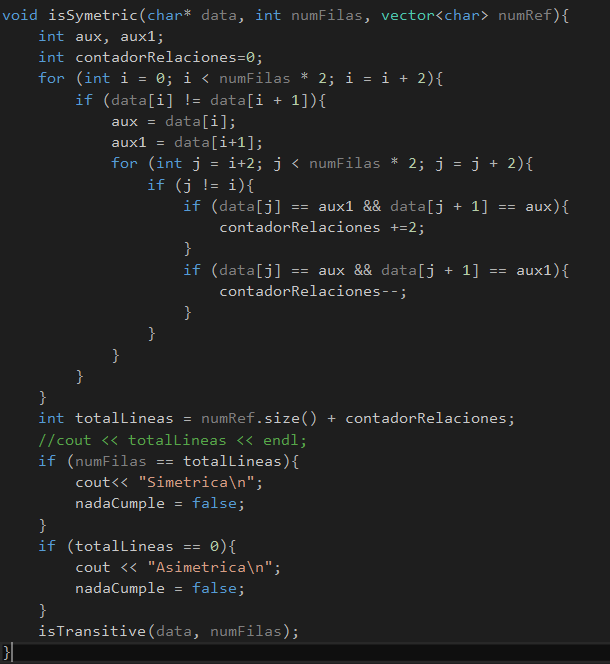
La función sort es una función donde le das un vector y te lo ordena entonces pasé los datos del arreglo al vector y después los ordené. Para quitar los repetidos se usó una función llamada unique e erase en donde si hay un dato repetido en el vector lo borra, por lo que la parte de mostrar los elementos del conjunto fue muy sencillo. Después de eso viene lo interesante.

1. La función para checar si las relaciones son reflexivas se llama is reflexive como se muestra aquí:



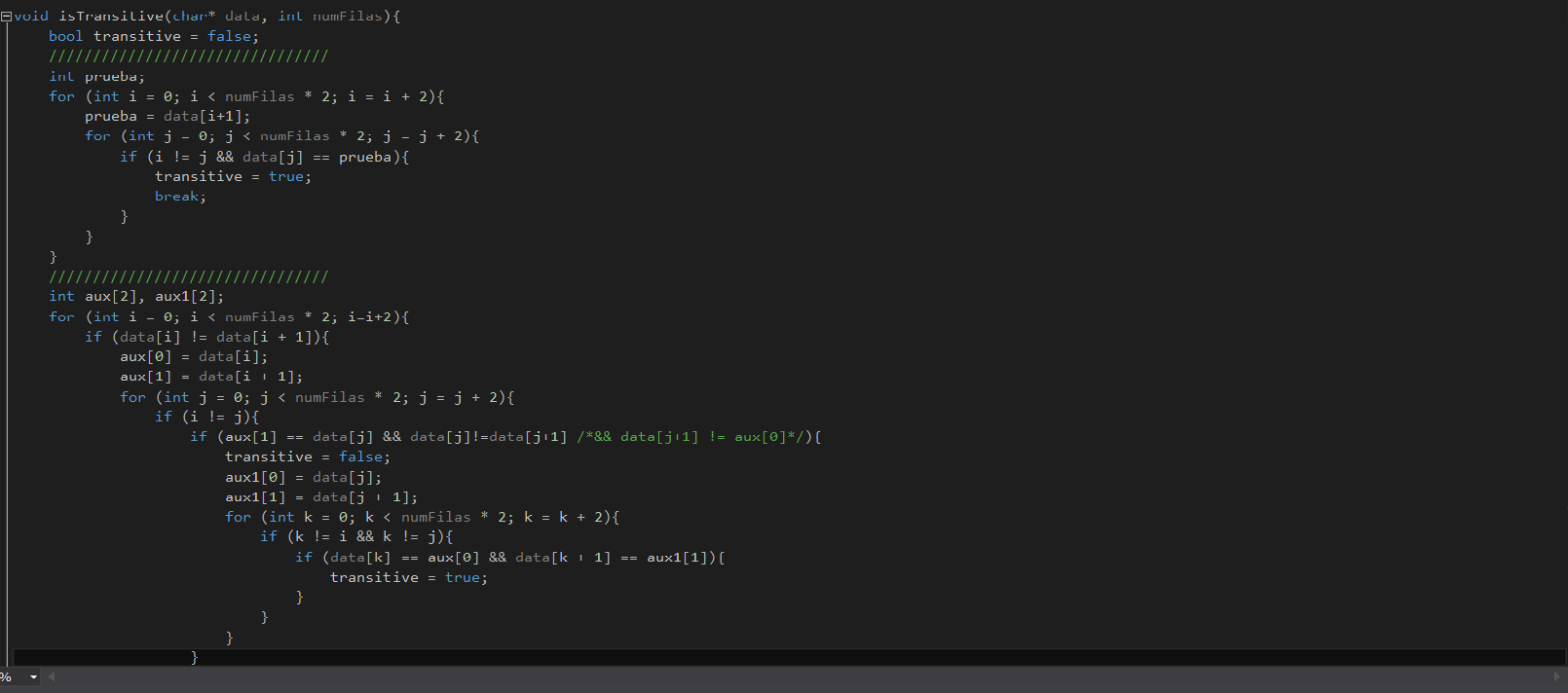
Y lo que hace esta función es crear un nuevo vector llamado ayuda y sacar el tamaño del vector en donde se imprimieron los elementos del conjunto. Se recorre y se busca relaciones de elementos con el mismo nombre por ejemplo una línea que sea: 1 1, si se encuentra se agrega un 1 al vector lo que indica que se relaciona consigo mismo, al finalizar solo uso la misma línea de código que use anteriormente para quitar los repetidos y comparo el número que hay con el número de elementos en el conjunto. Si coincide quiere decir que es reflexiva, y si el número te da 0 es que es irreflexiva.

1. La función para checar si las relaciones son simétricas se llama isSymetric y es la siguiente:



En esta función lo que se hace es ciclar dos, uno que contiene el primer par de números que se desea verificar si están pero de forma inversa y el otro que revisa si están de forma inversa, si sí se encuentran al revés se le suma un 2 al total de líneas ya que el segundo for no recorre desde el principio, si no recorre desde el donde está el contador del primer for. Por lo tanto el punto de esto es que si es simétrica tiene que tener todas las líneas de las relaciones excepto las que se relacionan cosigo mismo por ejemplo 1 1, por lo que al final se le suman el número de líneas que se relaciona consigo misma y eso debería de dar el total de líneas que introduce el usuario. Entonces tan solo se compara con el número de filas que introdujo el usuario y se determina si es simétrica. Ahora si el total de líneas te da 0 significa que es asimétrica.

1. La función para verificar si las relación es transitiva es:



Lo que determina si esta función es verdadera es que se tiene un booleano que inicia en falso, primero que nada se recorre el arreglo con 2 fors ciclados, el primero toma los valores de cada relación y el segundo lo que hace es verificar si hay una relación en la que el segundo número de la relación guardada en el primer for coincida con el primer número de la segunda relación, con que logre encontrar uno vuelve bool true y sigue su camino a los siguientes fors. Si no se encuentra entonces no es transitiva la relación.

El siguiente paso tiene 3 fors anidados. El primero (como el anterior) almacena relaciones que no sean de un elemento al mismo elemento. Se guardan estos datos y en el siguiente for se busca lo mismo que en el segundo for pasado, buscar que el segundo número del primer for sea igual al primer número del segundo for. Si esto se encuentra se busca en un tercer for que el primer número del primer for y el segundo número del segundo for estén relacionados. Si esto es verdad se sigue buscando hasta que cumpla para todos los casos. Lo que se busca es que tan solo uno de estos casos no se cumpla para que se diga que la relación no es transitiva.

Bibliografía:

Stackoverflow.com