Se pide que realice un programa en C++ que dado una o dos estructuras de datos (Tomando en cuenta que las operaciones con más de una estructura solo se realizan sobre el mismo tipo de estructura) sea capaz de:

* Concatenar dos estructuras en una sola: donde se debe crear una nueva estructura del tipo especificado que contenga los datos de la primera estructura especificada seguido de los datos de la segunda estructura especificada (Se debe conservar el orden de las estructuras originales en la estructura concatenada).
* Convertir una estructura en otra: donde se debe crear una nueva estructura del tipo especificado con los datos de la estructura especificada (Conservando el orden de la estructura original).
* Mezclar dos estructuras en una sola: donde se debe crear una nueva estructura del tipo especificado que contenga de manera alternada los elementos de las otras dos estructuras especificadas(Es decir un elemento de una seguido de un elemento de la otra y así sucesivamente, en caso de que las estructuras no tengan el mismo tamaño, cuando se terminen de insertar los elementos de una, se insertaran los elementos sobrantes de la otra al final de la estructura mezclada).
* Ordenar una estructura: donde se debe ordenar de forma ascendente la estructura especificada y convertirla al tipo de estructura especificada.
* Revertir el orden de una estructura: donde se debe revertir por completo el orden de la estructura especificada creando una nueva estructura del tipo especificado con el orden inverso de la original.

Para esto contara con un archivo de texto llamado estructuras.txt que contiene para cada caso de prueba lo siguiente:

* En la primera línea la cantidad de estructuras (1 o 2) para el caso de prueba.
* En la siguiente línea el/los tipo(s) de estructuras, el tipo de dato (numérico, carácter o cadena) y el tamaño separado por una coma (En caso de ser una sola estructura se especifica un solo tamaño, pero en caso de ser dos para el caso de prueba se especificaran el tamaño respectivo de cada una de las estructuras separado por una coma).
* En la siguiente línea los datos de la estructura separados por coma (en caso de ser dos estructuras sería una línea para cada estructura).
* En la última línea el tipo de operación a realizar y el tipo de estructura resultante de la operación separados por una coma.

Para cada caso de prueba se deben crear las estructuras especificadas, realizar la operación especificada y mostrar la estructura resultante en pantalla (Para realizar las operaciones se deben usar una o varias estructuras auxiliares, ya sean pilas, colas o listas).

Nota: Cuando se habla de conservar el orden de las estructuras originales luego de las operaciones, quiere decir que el tope de una pila, el frente de una cola o la cabecera de una lista deben comenzar con el mismo elemento para que se conserve el orden, por ejemplo, si tengo una cola(2,4,7,9) donde el frente es 2 y la llevo a una pila(2,4,7,9) el tope debe ser 2, y si lo llevo a una lista(2,4,7,9) la cabecera debe ser 2, así se conserva el orden de la estructura original. Al momento de las impresiones en pantalla, se debe imprimir de izquierda a derecha las estructuras, comenzando siempre con el tope, el frente o la cabecera.

Por ejemplo:

estructuras.txt

1

pila,numérico,6

5,20,8,9,11,2

convertir,cola

2

cola,carácter,6,5

c,r,t,j,t,y

f,d,j,u,i

concatenar,lista

2

lista,cadena,3,4

juan,jose,carlos

daniel,ramirez,juan,daniel

mezclar,pila

1

pila,cadena,4

carro,casa,pan,tenedor

ordenar,pila

1

cola,numérico,6

76,908,75,15,12,17

revertir,lista

Salida en pantalla:

Caso1(pila->convertir->cola):

5,20,8,9,11,2

Caso2(cola->concatenar->lista):

c,r,t,j,t,y,f,d,j,u,i

Caso3(lista->mezclar->pila):

juan,daniel,jose,ramirez,carlos,juan,daniel

Caso4(pila->ordenar->pila):

casa,carro,pan,tenedor

Caso5(cola->revertir->lista):

17,12,15,75,908,76

* Baremo de corrección:

Archivo diferente al del ejemplo con 10 casos de prueba, 1 punto por cada caso de prueba.