Pablo de la Mora Vega

A01020365

Reporte

Proyecto 2

**Manual de Usuario:**

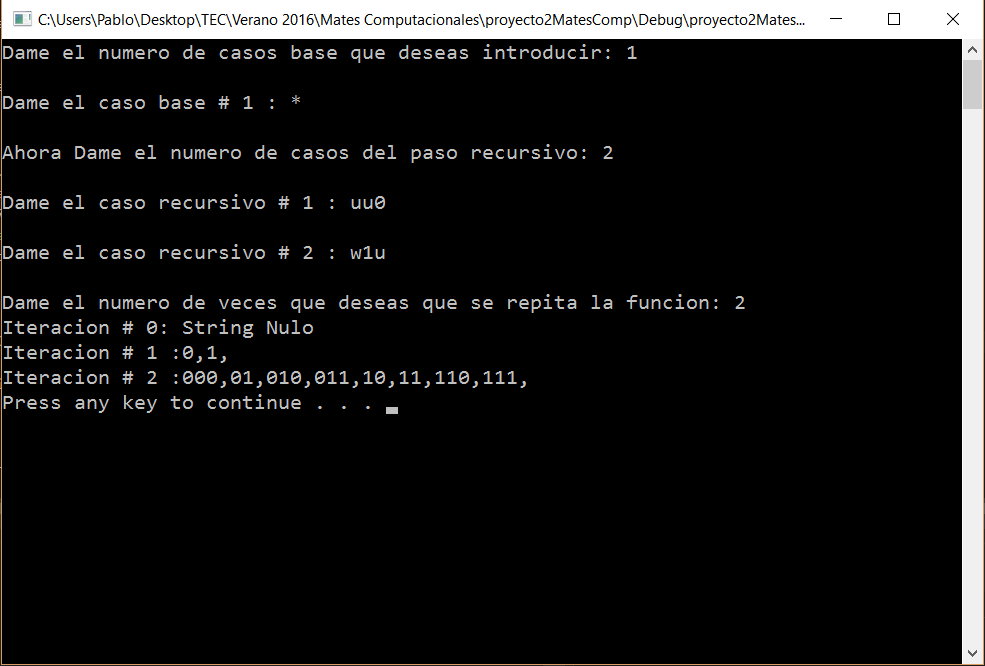
Este es un programa programado en C++. El programa no requiere de archivos de texto para poder correr, tan solo se corre y te va pidiendo que introduzcas los elementos necesarios para poder correrlo. El programa no valida la información que se introduce por lo que supone que los datos que se están introduciendo son válidos.

Primero el programa te pide que le des el número de casos base que deseas introducir, después de esto te pide que introduzcas los casos base (estos casos solo pueden contener un carácter). Ya que tienes esto ahora te pide que introduzcas el número de casos recursivos y después dichos casos recursivos.

Finalmente te pide que le digas cuantas llamadas a la función quieres que haga y el resultado te va a mostrar los strings creados en cada paso.

Para introducir el string nulo se tiene que poner con un “ \* ”. El asterisco es detectado y tomado como string nulo, si se introduce en los casos base.

A continuación muestro un ejemplo de los datos que se introducen para que el programa funcione correctamente:



En la imagen vemos que el usuario introdujo un \* por lo que es el string nulo y sus casos recursivos son: uu0 y w1u. El resultado se muestra por iteración, comenzando por la 0 que son los casos base introducidos. Después se repite el ciclo las veces que lo hayas indicado.

**Descripción Técnica del Código:**

Este código está programado en C++. Para almacenar los datos que se iban creando mientras corre el programa se usó la librería de vector y la librería de algorithm se usó para ordenar los datos y lograr borrar los repetidos.

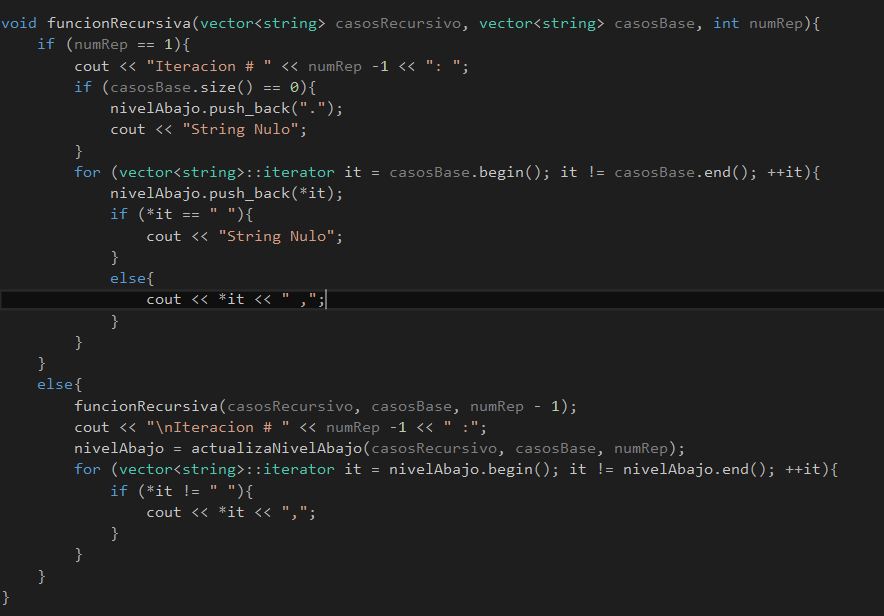
La función sort es una función donde le das un vector y te lo ordena entonces pasé los datos del arreglo al vector y después los ordené. Para quitar los repetidos se usó una función llamada unique e erase en donde si hay un dato repetido en el vector lo borra, por lo que la parte de mostrar los elementos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

Ahora bien la parte interesante del proyecto está basada en dos funciones. Una llamada funcionRecursiva y la segunda llamada actualizaNivelAbajo.

La función funcionRecursiva recibe los parámetros de casosRecursivo, casosBase y número de repeticiones.

Después de ser llamada en el main ya que se tiene la información necesaria para que pueda funcionar bien lo que hace es ver el número de repeticiones actual, si el número no es igual a 1 entonces es donde entra la parte recursiva. La función se llama a si misma hasta que llegue a su caso base, que a su vez es el nivel más bajo donde se encuentran los casos base.

En esta imagen se muestra la implementación:



Ahí se muestra cómo se va llamando a si misma hasta llegar a su caso base de donde parte de regreso. Cuando llega caso base lo que se hace es que hay un vector llamado nivelAbajo que es de donde se van sacando los nuevos niveles, entonces al llegar al caso base el nivelAbajo se le agregan los elementos base que introdujo el usuario.

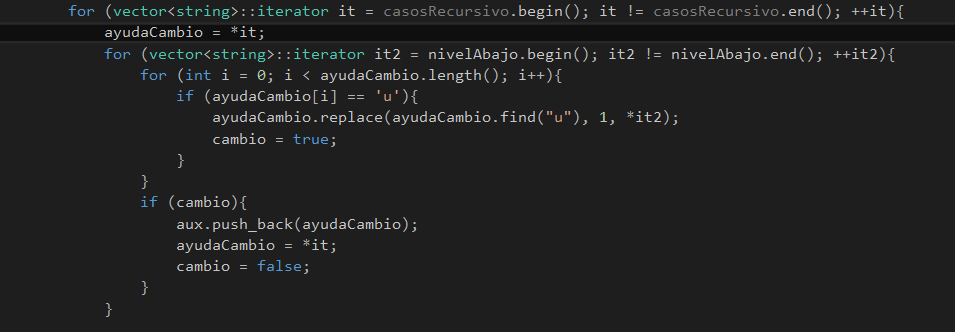
Ya que se hizo esto pasa al siguiente nivel en la recursión y se llama a la segunda función importante de este programa. La función actualizaNivelAbajo se llama y regresa un vector que obtendrá el vector nivelAbajo actualizado. La función recibe los mismos argumentos que funcionRecursiva, 2 vectores y un entero.

Lo demás de la función funcionRecursiva se usa para imprimir lo necesario.

Funcion actualizaNivelAbajo:

Esta función primero que nada va a ir recorriendo el vector de casos recursivos con un for, y adentro se van a implementar varios fors para recorrer el nivel de abajo que contienen ya strings aceptados por el lenguaje. Lo que va a hacer es recorrer el nivel de abajo y va a buscar una de las variables que representen los strings. Cada letra que representa un string tiene su for de nivelAbajo para poder checar las letras e intercambiarlas por todos los strings del nivelAbajo.

A continuación muestro el ciclo para la u pero este ciclo se repite para todas las letras que representan strings:

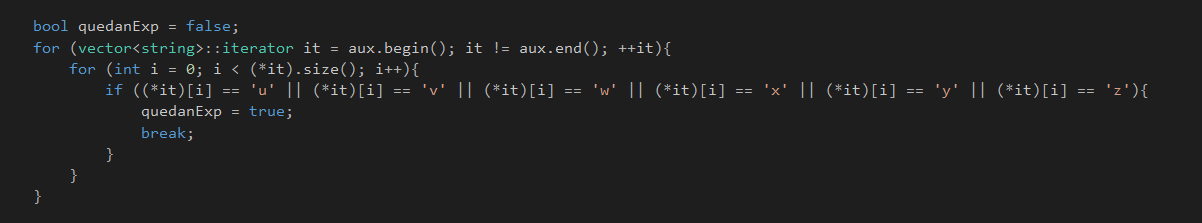


Ya que encuentra una letra de las que busca lo que hace es que le hace un push con el cambio al vector aux y regresa la original a como estaba para seguir buscando las demás.

Haciendo esto lo que garantizas es que en el vector aux vas a tener sustituido las letras que representan strings pero con esta estrategia en aux va a tener agregados variables aax, en el caso recursivo uax. Esto quiere decir que si se tienen 2 variables de strings, en el vector aux van a seguir agregados strings que aún se tienen que cambiar.

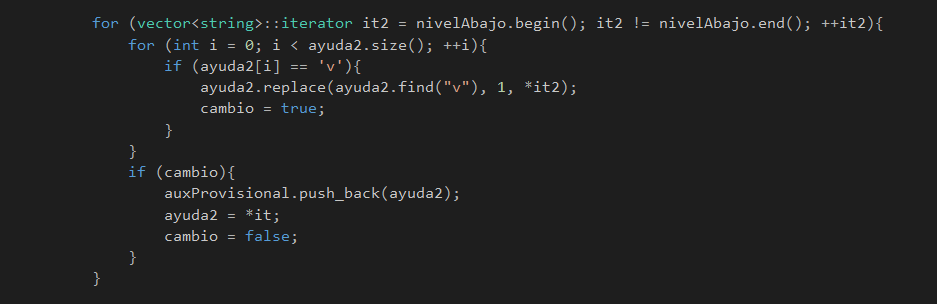
Sabiendo esto entonces pasando esta parte se hace una revisión de los strings de aux en busca de alguno de estos caracteres. Si se encuentra activa un booleano que a su vez es la condición de un while. Mientras encuentre uno de estos elementos el while continuara llamándose.

Lo que se hace en este paso es muy parecido a lo que se hizo anteriormente con la diferencia es que se toman los elementos del vector y se analizan. Si no contiene uno de los caracteres que estamos buscando entonces se introduce al nuevo vector de ayuda. Esto se hace en este sencillo paso:



Ahí se valida si todavía contiene alguno de los caracteres que estamos buscando.

Ahora bien, si encuentra uno de esos caracteres lo que se hace es lo mismo que se hizo en el primero paso de esta función. Remplazarlo con nivelAbajo y regresarlo a su valor original. Lo que va a ocasionar esto es que se tenga cada vez más strings hasta que se hayan eliminado todos los u,v,w,x,y,z.

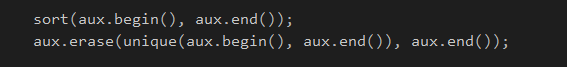


En esa imagen ahora se muestra la búsqueda de v, pero repito que es el mismo ciclo para todas las letras.

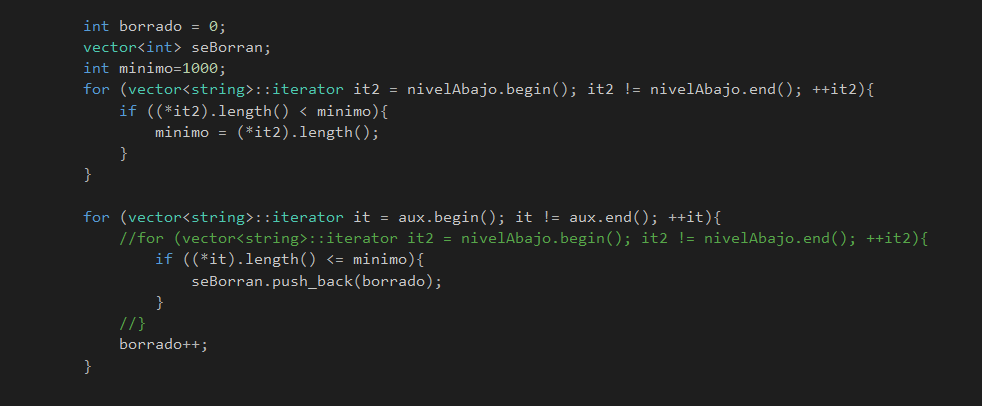
Al final vamos a tener una gran cantidad de strings en el vector.

Lo que se hace ahora es empezar a ver el caso de string nulo. Si se contiene el string vacío se va a agregar un “ “ al vector para que se tome como elemento y después solamente se borra. Esto cumple la simulación del string nulo.

Ahora lo único que falta es eliminar los repetidos por lo que se utiliza el sort y el erase unique para limpar.



Lo último relevante de este código es que cuando hay string nulo y se intenta simular con el espacio entonces también te crea strings que están uno o dos niveles arriba por lo que se hace una validación de que el string no puede ser de tamaño igual o menor al menor del nivelAbajo de esta manera:



Con esto ya quedan los elementos que se desean y solo se necesitan imprimir para poder imprimirlos y mostrárselos al usuario.

Bibliografía:

Stackoverflow.com