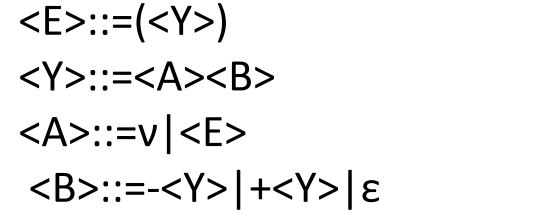
Δεύτερο θέμα

Στην δεύτερο θέμα έπρεπε να δημιουργηθεί μια γεννήτρια συμβολοσειρών η οποία να υπακούει στους παρακάτω γραμματικούς κανόνες:



Γλώσσα προγραμματισμού : C++

Βάση θεωρίας πρέπει να δημιουργείται μια συμβολοσειρά η οποία να ξεκινάει από το <Ε> και να καταλήγει σε μια αλληλουχία από σύμβολα όπως το ν , οι παρενθέσεις ( ) , καθώς και το κενό διάστημα .Όπως βλέπουμε στους κανόνες για το <Α> και <Β> έχουμε περισσότερες από μια επιλογές στην αντικατάσταση του συμβόλου επομένως πρέπει το πρόγραμμα σε αυτό το σημείο να κάνει την επιλογή μόνο του κατά τύχη. Παράλληλα όντας γεννήτρια το πρόγραμμα δεν απαιτεί καμία είσοδο δεδομένων από τον χρήστη πέρα από την εκκίνηση του προγράμματος .Τέλος με αυτήν την τυχαία επιλογή δεν εξασφαλίζεται ότι το πρόγραμμα θα τελειώσει ,γι΄αυτο τον λόγο το η συνάρτηση η οποία αντικαθιστά τα σύμβολα σταματά μετά το πέρας των 50 επαναλήψεων, και άμα έχουν γίνει επιλογές τέτοιες ώστε να παραμένουν μη τερματικά σύμβολα αυτά θα παραμείνουν στο τέλος της συμβολοσειράς καθώς αν τα αλλάζαμε θα προσβάλουμε την αυτήν την τυχαία επιλογή που κάνει το πρόγραμμα.

Τα παραπάνω υλοποιούνται ως εξής :

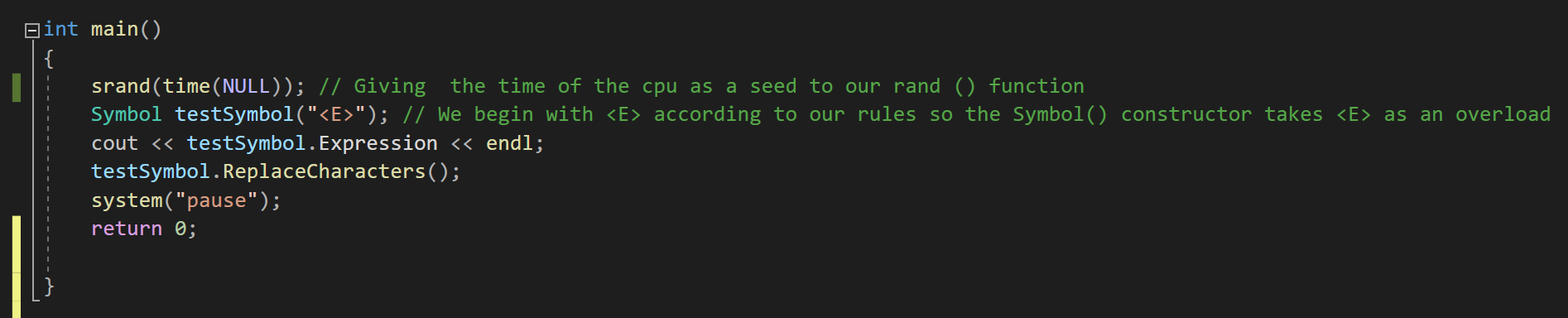
Για να γίνει εφικτή η επεξεργασία συμβολοσειρών πρέπει να χρησιμοποιηθεί η βιβλιοθήκη string :



Για να γίνεται η τυχαία επιλογή πρέπει να χρησιμοποιηθεί η βιβλιοθήκη ctime:



Αυτή η βιβλιοθήκη διαθέτει την συνάρτηση rand ( ) η οποία με την αρχικοποίηση της από το ρολόι του επεξεργαστή μπορεί να δώσει τυχαίους ακέραιους αριθμούς από το ένα μέχρι και ένα ανώτατο φράγμα που θα τεθεί από τον δημιουργό το προγράμματος.



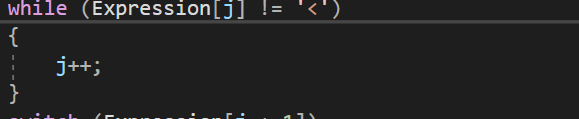
Για μεγαλύτερη ευκολία έχει δημιουργηθεί μια τάξη με το όνομα Symbol.

Στης οποίας τον constructor μπαίνει το σύμβολο εκκίνησης που είναι το <Ε>



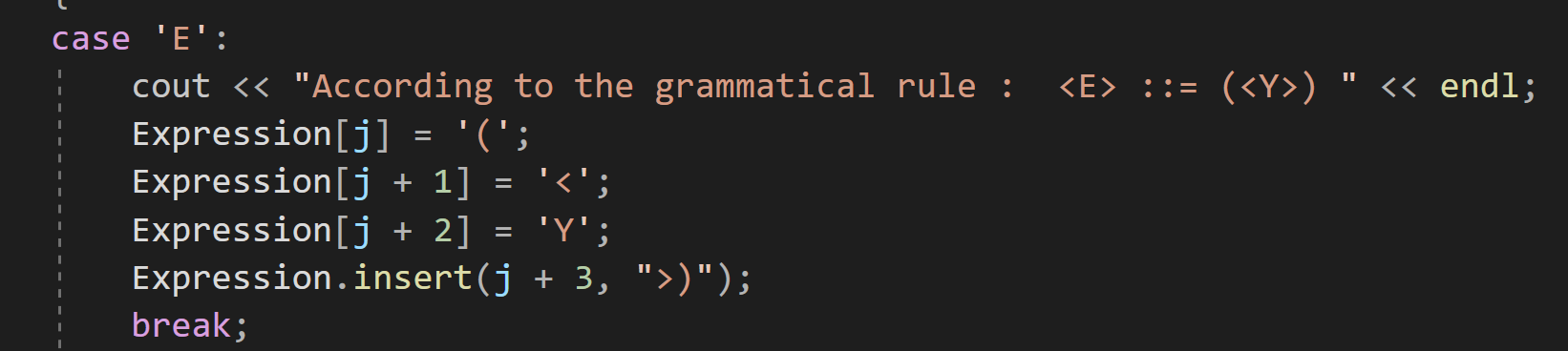
Κατόπιν, καλείται η συνάρτηση ReplaceCharacters() η οποία :

Αρχικά ελέγχει το αριστερότερο σύμβολο για το αν είναι τερματικό η όχι. Αυτό επιτυγχάνεται με την συνάρτηση Checker ( ) ή οποία ελέγχει το αριστερότερο μέλος του πεδίου Expression ,το οποίο είναι η συμβολοσειρά που θα έχει ως έξοδο το πρόγραμμα. Η Checker ( ) ελέγχει αν το Expression [0] είναι ίσο με τον χαρακτήρα ‘<’ που υποδεικνύει την ύπαρξη μη τερματικού συμβόλου , ειδάλλως προχωρά στο αμέσως επόμενο χαρακτήρα ώστε να βρει το επόμενο μη τερματικό :

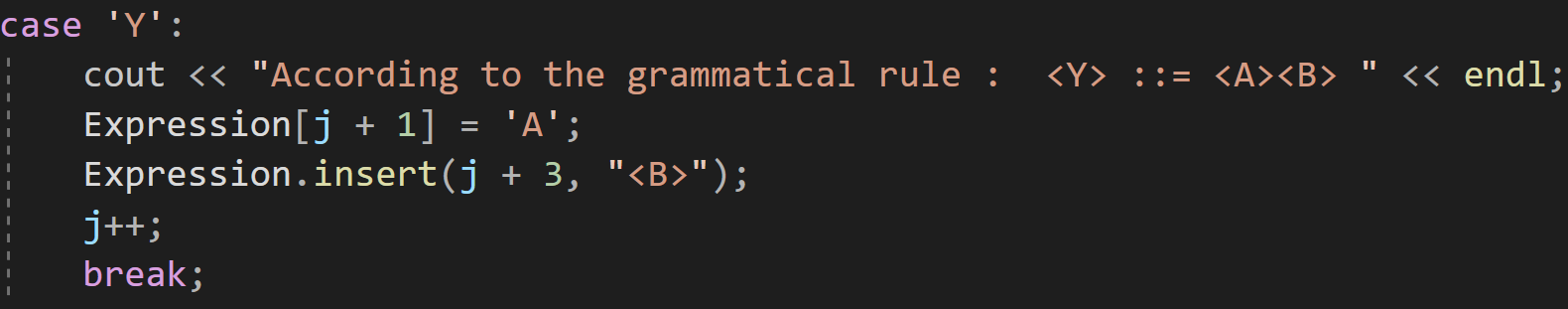


Μετά οι ReplaceCharacters( ) αλλάζει τα σύμβολα , τα πιθανά σενάρια που μπορούμε να έχουμε είναι τα εξής :

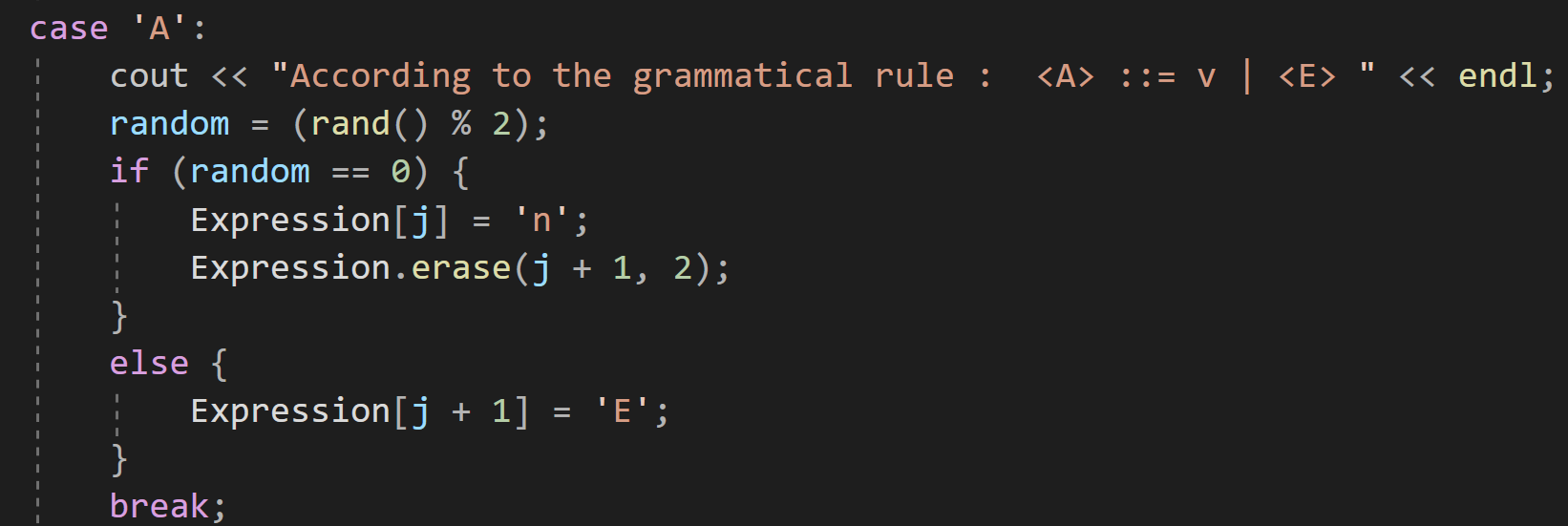
1. Το πιο αριστερό σύμβολο να είναι <E> εκεί θα προστεθεί το σύμβολο (<Υ>) :

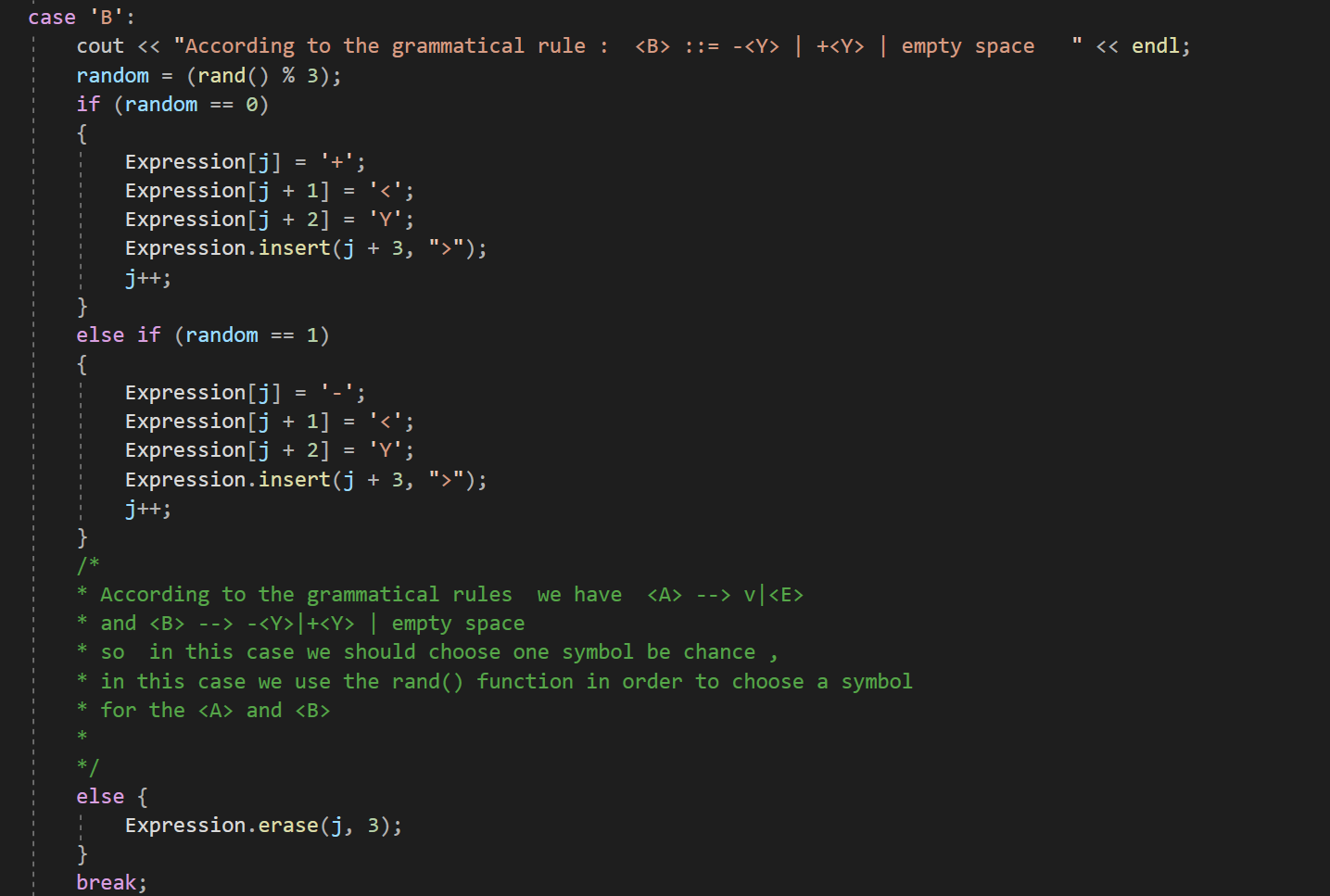


2. Το πιο αριστερό σύμβολο να είναι <Υ> εκεί θα προστεθούν τα σύμβολα <Α><Β> :



3. Το πιο αριστερό σύμβολο να είναι <Α> εκεί θα προστεθούν τα σύμβολα ν ή <Ε> ανάλογα το output της random ( ):



4. Tο πιο αριστερό σύμβολο <Β> εκεί θα προστεθούν τα σύμβολα -<Υ>| +<Υ> |ε : 

Η παραπάνω διαδικασία επαναλαμβάνεται για έως και 50 φορές αν την πεντηκοστή φορά δεν έχουν αλλάξει όλα τα σύμβολα σε μη τερματικά τότε όπως ειπώθηκε παραπάνω , παραμένουν όσα μη τερματικά σύμβολα. Τέλος στις περιπτώσεις όπου έχουμε να αντικαταστήσουμε το <Α> ή το <Β> τότε η rand ( ) μας δημιουργεί ακέραιους αριθμούς στο διάστημα [0,2) για το <Α> και

στο διάστημα [0,3) για το <Β> ,οι οποίοι αντιστοιχούν σε μια από τις δυνατές επιλογές αντικατάστασης .

Τέλος έχουμε και την συνάρτηση Main ( ) στην οποία γίνεται αρχικοποίηση του συμβόλου εκκίνησης ως αντικείμενο της τάξης Symbol ( ) :



Και με το στιγμιότυπο testSymbol καλούμε την συνάρτηση ReplaceCharacters ( ) που μας αντικαθιστά παράγει την συμβολοσειρά .