Τίτλος Ασκησης:	Εισαγωγή στην λειτουργία λεκτικών αναλυτών
Εργαστήριο:	1
Απαραίτητα Εργαλεία:	Visual Studio Professional 2008 (προτεινόμενο) /2010/2012
Απαιτούμενες Γνώσεις:	C/C++
Στόχοι:	1) Εξοικείωση με την λειτουργία του λεκτικού αναλυτή 2) Εξοικείωση με το περιβάλλον του Visual Studio
Διορία Παράδοσης:	Εντός της διδακτική ώρας του **<u>τρέχοντος εργαστηρίου</u>**
Παραδοτέα:	1) Επιδειξη λειτουργίας και Πηγαία Αρχεία (5/10) 2) Γραπτή αναφορά (στο χαρτί) των ενεργειών που έγιναν (5/10)
Διδάσκων:	Γρηγόρης Δημητρουλάκος

Στόχος εργαστηριακής ασκησης : Στην παρούσα άσκηση ο φοιτητής καλείται να φτιάξει έναν στοιχειώδη λεκτικό αναλυτή που διαθέτει ταυτόσιμα χαρακτηριστικά με τους λεκτικούς αναλυτές που παράγει το εργαλείο flex. Με αυτόν τον τρόπο ο φοιτητής αντιμετωπίζει ο ίδιος τις σχεδιαστικές ανάγκες που ανακύπτουν ώστε στην συνέχεια να γίνει καλύτερα η εμπέδωση των λειτουργιών του λεκτικού αναλυτή που παράγεται από το flex.

Επιπλέον η παρούσα άσκηση αναδεικνύει την χρησιμότητα του εργαλείου flex όπου ο χρόνος υλοποίησης της άσκησης 1 με χρήση του flex για παραγωγή του ίδιου λεκτικού αναλυτή είναι σημαντικά μικρότερος.

Ασκηση 1: Δημιουργία ενός λεκτικού αναλυτή σε γλώσσα C που θα βρίσκει **αλφαριθμητικά και αριθμούς** από αρχεία κειμένου και θα επιστρέφει **για ένα κάθε φορά τον τύπο και την σημασιολογική του τιμή** στην καλούσα υπορουτίνα

α) Δημιουργία ενός C/C++ project στο Visual Studio

b) Δημιουργία Συνάρτησης `yylex`

c) Δημιουργία συνάρτησης `main`

d) Ορισμός της union SEMANTIC INFO στο πάνω μέρος του αρχείου μετά τα includes

e) Υλοποίηση συνάρτησης `yylex` ώστε κατά την κλήση να επιστρέφει ένα αλφαριθμητικό που θα βρεί από το αρχείου κειμένου στην καλούσα υπορουτίνα. Κατά τις υπόλοιπες κλήσεις θα επιστρέφει τα υπόλοιπα αλφαριθμητικά αν υπάρχουν ή το τέλος του αρχείου. Το πρότυπο της συνάρτησης yylex() θα είναι

```
int yylex(FILE *fpointer, union semantic_info *sem);
```

όπου *sem* είναι ένας δείκτης προς την σημασιολογική τιμή της λεκτικής μονάδας που έχει ταυτοποιηθεί και η επιστρεφόμενη τιμή αν είναι 1 δηλώνει ότι έχει βρεθεί αλφαριθμητικό, 2 ότι έχει βρεθεί αριθμός, ενώ αν είναι 0 σημαίνει ότι έχει φτάσει στο τέλος του αρχείου.

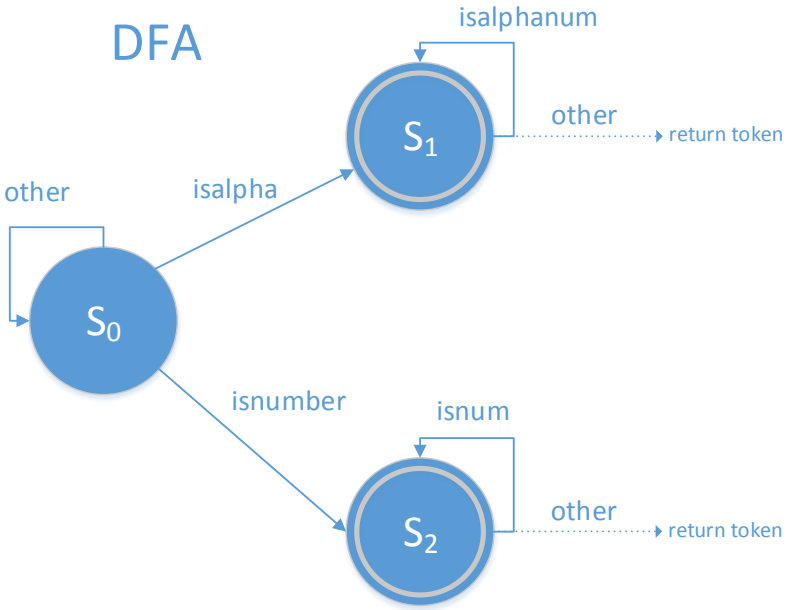
f) Υλοποίηση της συνάρτησης `main()` όπου θα ανήγει ένα αρχείο κειμένου που θα δίνει ο χρήστης από την γραμμή εντολής και θα καλεί την `yylex()` επαναληπτικά μέχρι να βρεθεί το τέλος του αρχείου. Κάθε φορά που δέχεται μια λεκτική μονάδα θα την τυπώνει στην οθόνη μαζί με τον τύπο της. Ενώ στο τέλος θα τυπώνει και το πλήθος των λεκτικών μονάδων από κάθε τύπο (αριθμοί και αλφαριθμητικά) που βρέθηκαν.

Υποδείξεις :

1) **Βοηθητικές Συναρτήσεις:** `fopen, fclose, isalnum, isalpha, isdigit, fscanf, fgetc, fgets, strtok,strspn, strcspn`
2) **Αλφαριθμητικό** : Είναι μια λεκτική μονάδα που ο πρώτος χαρακτήρας είναι γράμμα και οι υπολοιποι αν υπάρχουν είναι αριθμοί ή γράμματα
3) **Αριθμοί** : Είναι λεκτικές μονάδες που αποτελούνται από αριθμούς
4) Η **σημασιολογική τιμή της λεκτικής μονάδας** αποθηκεύενται στην παρακάτω δομή δεδομένων. Για την περίπτωση του αριθμού η σημασιολογική τιμή είναι ένας ακέραιος (int i) ενώ για το αλφαριθμητικό ένας πίνακας από χαρακτήρες (char * s). Οταν αναγνωριστεί ένας αριθμός ή ένα αλφαριθμητικό η τιμή του εξάγεται με κατάλληλο τρόπο από την λεκτική μονάδα και αποθηκεύται στην μεταβλητή SEMANTIC_INFO.

```
union semantic_info{
    char *s;
    int i;
} SEMANTIC_INFO;
```

5) Υλοποίηση **μηχανής κατάστασης**



6) Υλοποίηση του DFA με την μέθοδο direct-coded

```
switch (state){
        case 0: // state0
                if ( nextcharacter == isalpha ){
                        Action1;
                }
                else if ( nextcharacter == isnumber ){
                        Action2;
                }
                else {
                        Action3;
                }
        case 1: // state1
        ...
        ...
```

7) Δομή συνάρτησης main()

```
void main(...){
...
int tokentype;
        while ( (tokentype = yylex(fp,sem)) != 0 ){
                if ( tokentype == 1 ){
                        Action1;
                }
                else if ( tokentype == 2){
                        Action2;
                }

        }
...
}
```

8) Δομή συνάρτησης yylex()

```
int yylex(...){
int nextcharacter;
char lexeme[100];
        while ( (nextcharacter = fgetc(fp)) != EOF ){
                DFA;
        }

        AssempleSemanticInfo();

}
```

9) **Προτεινόμενα βήματα:** α) Εξοικείωση με τις ρουτίνες ανοίγματος και ανάγνωσης αρχείων. Επιλογή των κατάλληλων συναρτήσεων που θα βοηθήσουν στον σκοπό της άσκησης β) Υλοποίηση της μηχανής κατάστασης η οποία θα δέχεται χαρακτήρες και θα μεταβαίνει στην ανάλογη κατάσταση και γ) Υλοποίηση των κατάλληλων ενεργειών ανάλογα με την κατάσταση στην οποία έχει επέλθει η μηχανή κατάστασης

Περιεχόμενα Γραπτής Αναφοράς

1) Περιγραφή των βημάτων a,b,c
2) Δηλώσεις των απαράιτητων μεταβλητών που χρησημοποιήθηκαν
3) Δομές ελέγχου και συνθήκες που χρησιμοποιήθηκαν για την σάρωση του αρχείου
4) Με ποια στρατηγική διαβάστηκε το αρχείο εισόδου
5) Ποιές βοηθητικές συναρτησεις και με ποίο τρόπο χρησημοποιήθηκαν για την αναγνώριση αλφαριθμητικών και αριθμών
6) Αλγοριθμικό διάγραμμα ροής