

Τίτλος Άσκησης:	Κανονικές Εκφράσεις
Εργαστήριο:	4
Απαραίτητα Εργαλεία:	Visual Studio Professional 2008, Flex για windows v2.5.4a, flexbisonrules (http://msdn.microsoft.com/en-us/library/aa730877(v=vs.80).aspx)
Απαιτούμενες Γνώσεις:	C/C++
Βιβλιογραφία:	1) Flex & Bison του OReilly 2009 (Ξενόγλωσσο)
Στόχοι:	1) Εξοικείωση με την συγγραφή κανονικών εκφράσεων
Διορία Παράδοσης:	Εντός της διδακτικής ώρας του τρέχοντος εργαστηρίου
Παραδοτέα:	1) Επίδειξη λειτουργίας και Πηγαία Αρχεία (5/10) 2) Γραπτή και προφορική εξέταση στις ερωτήσεις που τίθενται (5/10)
Διδάσκων:	Γρηγόρης Δημητρουλάκος
Απορίες:	dhmhgre@uop.gr και καθημερινά στο εργαστήριο λογικής σχεδίασης

Στόχος εργαστηριακής άσκησης : Στην παρούσα άσκηση ο φοιτητής μαθαίνει εξασκείται στην συγγραφή κανονικών εκφράσεων για λεκτικούς αναλυτές flex. Οι λεκτικοί αναλυτές flex διαθέτουν για την αναγνώριση προτύπων εκτός από τις κανονικές εκφράσεις και μια διεπαφή χρήση σε γλώσσα C/C++ μέσω της οποίας η αναγνώριση προτύπων μπορεί να υποστηριχθεί. Επιπλέον οι κανονικές εκφράσεις διαθέτουν την κατάλληλη σημασιολογία για την αναγνώριση προτύπων με συμφραζόμενα.

Προ-απαιτούμενες Ενέργειες:

- 1) Εγκατάσταση των εργαλείων flex, flexbisonrules σύμφωνα με τις οδηγίες που έχουν αναρτηθεί στο eclass (Στο εργαστήριο υπάρχουν)
- 2) Διάβασμα υποδείξεων – Προετοιμασία στο σπίτι

Άσκηση 1: Σας δίνεται το ακόλουθο ημιτελές πρόγραμμα:

```
%option noyywrap
%{
#include <stdio.h>
#include <stdlib.h>
FILE *fout;
%}
%%
// Write Regular Expressions Here ...
%%

void main( int argc, char ** argv){
    if ( argc < 3 ){
        printf("\nError!!! Missing Command line arguments");
        printf("\nUsage exe <inputfile> <outputfile>");
        exit(1);
    }
    else{
        fout = fopen(argv[2], "w");
        yyin = fopen(argv[1], "r");
        yylex();
    }
}
```

- 1) **Συγγραφή Κανονικών Εκφράσεων:** Να γραφούν οι κατάλληλες κανονικές εκφράσεις για την αναγνώριση των παρακάτω προτύπων και να γίνει επαλήθευση με τα παραδείγματα που δίνονται για κάθε ερώτημα. (Για την επαλήθευση τοποθετήστε τα παραδείγματα στο αρχείο εισόδου και επιβεβαιώστε ότι δεν εμφανίζεται τίποτα στο stdout με την εκτέλεση του λεκτικού αναλυτή) (**Να γίνει πλήρης εξήγηση των απαντήσεων που δίνετε**)
- a. **String Literals** δηλαδή αλφαριθμητικά που εμφανίζονται εντός ζεύγους διπλών (" ") ή μονών (' ') εισαγωγικών. (πχ printf("\This is a string embedded in a string\"); Για την επαλήθευση stdout γράψτε την κατάλληλη ενέργεια για να τυπωθεί το γγtext στο αρχείο εξόδου.
 - b. **Σχόλια** γραμμής (//) και σχόλια πολλαπλών γραμμών (/* */). Εδώ πρέπει να εξαιρεθούν σχόλια που βρίσκονται μέσα σε αλφαριθμητικά (πχ " /* Its not a comment */ ")
Προτείνεται η χρήση της συνάρτησης (input/output) για την εν μέρει επίλυση του ερωτήματος
 - c. **Ακέραιοι αριθμοί** (πχ -1, 2, -34, +45)
 - d. **Δεκαδικοί αριθμοί** (πχ 0.1, .34, -.45, -1.22, +23.34, -23. , +.345, -.45)
 - e. **Αριθμοί κινητής υποδιαστολής** (πχ 4.563E+11, 1.56e+11, 1.56X10^11, +1.56x10^11, 1.34X10*11, -1.56x10*11, 1.0E-6, 1.0e-6, +1.0X10^-6, +1.0x10^-6, -1.0X10*-6, -1.0x10*-6)
 - f. **Κενά αυθαίρετου μεγέθους** (πχ *κενα + tabs * για την επαλήθευση πρέπει στο stdout να εμφανιστεί **)
 - g. **Τελεστές Πράξεων** (πχ ++, +, !=, ==, &&, &, /, *, **, ^, --, - για την επαλήθευση προσθέστε την κατάλληλη ενέργεια ώστε να τυπώνει στο stdout ποιον τελεστή ταυτοποίησε κάθε φορά)
 - h. **Ονόματα Αρχείων:** Τα ονόματα αρχείων θα εγκλείονται σε διπλά (") ή μονά (') εισαγωγικά θα ξεκινάνε από γράμμα και θα ακολουθούν αριθμοί, γράμματα ή underscore ακολουθούμενα από την τελεία (.) και την προέκταση που θα έχει από 1 μέχρι 3 χαρακτήρες γράμματα ή αριθμούς (Για την επαλήθευση χρησιμοποιήστε και ονόματα αρχείων που παραβιάζουν τον παραπάνω κανόνα πχ "flex.cred", "3flex.f", 'flex.', "weird.fil")
 - i. **Δεσμευμένες Λέξεις:** Σε όλες τις γλώσσες υπάρχουν δεσμευμένες λέξεις οι οποίες αναγνωρίζονται κατά το στάδιο της λεκτικής ανάλυσης και αποστέλλονται στον συντακτικό αναλυτή για περαιτέρω επεξεργασία. Γράψτε τις κανονικές εκφράσεις για 5 δεσμευμένες λέξεις της γλώσσας C της προτίμησής σας και επαληθεύστε την λειτουργία του λεκτικού αναλυτή σε συνδυασμό με άλλα αλφαριθμητικά.
 - j. **Αναγνώριση προτύπων με εκ δεξιών συμφραζόμενα:** Ο αριθμός μητρώου ενός φοιτητή έχει την μορφή cstSSSSYYYY όπου YYYY είναι το έτος εισαγωγής και SSSS ένας αύξων αριθμός. Να γραφτεί η κανονική έκφραση που θα αναγνωρίζει τους αριθμούς μητρώου φοιτητών (cstSSSS) με έτος εισαγωγής 2010. Η είσοδος θα είναι ο πλήρης αριθμός μητρώου (cstSSSSYYYY) και στο αρχείο εξόδου θα αναγράφονται οι μόνοι οι αριθμοί μητρώο κατά το έτος επιλογής (cstSSSS)
 - k. **Αναγνώριση προτύπων με εξ αριστερών συμφραζόμενα:** Να προσδιοριστούν και να εξαχθούν στο αρχείο εξόδου τα αλφαριθμητικά που ξεκινάνε από την αρχή των γραμμών του αρχείου εισόδου

Υποδείξεις :

Πρότυπο/Μοτίβο	Επεξήγηση
.	Ταιριάζει με οποιονδήποτε χαρακτήρα εκτός από την αλλαγή γραμμής (\n). Χρησιμεύει πολλές φορές για τον κατακερματισμό ενός αρχείου κειμένου στις γραμμές του
[]	Ταιριάζει με οποιονδήποτε χαρακτήρα που βρίσκεται μέσα στις αγκύλες. Αν ο πρώτος χαρακτήρας είναι ο ^ τότε το νόημα αλλάζει σε "Ταιριάζει με οποιονδήποτε χαρακτήρα που δεν βρίσκεται μέσα στις αγκύλες". Χρησιμοποιώντας τον χαρακτήρα - (παύλα) μπορεί να ορισθεί ένα εύρος χαρακτήρων π.χ [0-9] σημαίνει [0123456789], [a-z] σημαίνει οποιοδήποτε μικρό γράμμα, [A-Z] σημαίνει όλα τα κεφαλαία και μικρά γράμματα. Προσοχή σε αυτή η σημειογραφία λαμβάνεται υπόψη η σχετική τιμή που έχει κάθε χαρακτήρας στην κωδικοποίηση χαρακτήρων του συστήματος π.χ στον κώδικα ASCII τα κεφαλαία γράμματα έχουν μικρότερες τιμές από τα μικρά.
^	Ταιριάζει με την έναρξη της γραμμής σε ένα αρχείο κειμένου. πχ ^a σημαίνει ότι υπάρχει ταίριασμα όταν ο χαρακτήρας a βρίσκεται στην αρχή της γραμμής.
\$	Ταιριάζει με τον τελευταίο χαρακτήρα γραμμής σε ένα αρχείο κειμένου πχ a\$ σημαίνει ότι υπάρχει ταίριασμα όταν ο χαρακτήρας a είναι ο τελευταίος χαρακτήρας που εμφανίζεται σε μια γραμμή.
{ }	Εάν στις αγκύλες υπάρχουν ένας ή δύο αριθμοί τότε αυτό αντιστοιχεί στους αριθμούς επανάληψης του μοτίβου που βρίσκεται πριν την αγκύλη για να έχουμε ταίριασμα. π.χ A{1,3} ταιριάζει με τα αλφαριθμητικά A, AA, AAA και μόνον αυτά, A{2} ταιριάζει με το AA και μόνο. Εάν μέσα στις αγκύλες υπάρχει ένα όνομα τότε γίνεται αναφορά σε κάποιο μοτίβο που έχει ορισθεί με αυτό το όνομα στην περιοχή ορισμών (πρώτη περιοχή) πχ αν ορισθεί INTEGER [0-9]* ... τότε η έκφραση {INTEGER} ισοδυναμεί με το [0-9]*
\	Χρησιμοποιείται ως χαρακτήρας εξόδου για να προσδώσει σε έναν χαρακτήρα σημασιολογία ελέγχου. πχ \n αντιστοιχεί στην αλλαγή γραμμής
*	Ταιριάζει με αλφαριθμητικό που περιέχει <u>καμία</u> ή <u>περισσότερες</u> επαναλήψεις του μοτίβου που προηγείται του χαρακτήρα * πχ [0-9]* ταιριάζει με τον κενό χαρακτήρα ή οποιοδήποτε αριθμό με οσαδήποτε ψηφία.
+	Ταιριάζει με αλφαριθμητικό που περιέχει <u>μία</u> ή <u>περισσότερες</u> επαναλήψεις του μοτίβου που προηγείται του χαρακτήρα + πχ [0-9]+ ταιριάζει με οποιοδήποτε αριθμό με οσαδήποτε ψηφία.

?	Ταιριάζει με αλφαριθμητικό που περιέχει <u>μία</u> ή <u>καμία</u> επαναλήψεις του μοτίβου που προηγείται του χαρακτήρα * πχ [0-9]? ταιριάζει με τον κενό χαρακτήρα ή ένα οποιοδήποτε ψηφίο μεταξύ του 0 και 9.
	Ο τελεστής διάζευξης χρησιμοποιείται για το ταίριασμα μιας εκ των δύο κανονικών εκφράσεων που βρίσκονται εκατέρωθεν του χαρακτήρα . πχ up down ταιριάζει με τα 2 αλφαριθμητικά up ή down και μόνον αυτά
"..."	Ταιριάζει με οποιοδήποτε αλφαριθμητικό που είναι ίσο με το αλφαριθμητικό που βρίσκεται μέσα στα εισαγωγικά πχ "john" ταιριάζει μόνο με το αλφαριθμητικό john
()	Οι παρενθέσεις χρησιμοποιούνται για την σύνθεση πολυπλοκότερων κανονικών εκφράσεων από απλούστερες
/	Αναγνώριση με συμφραζόμενα. Σε αυτή την περίπτωση υπάρχει ταίριασμα ενός αλφαριθμητικού με την κανονική έκφραση στα αριστερά της κάθετου αν και μόνο αν το αλφαριθμητικό ακολουθείται από δευτερεύον αλφαριθμητικό που ταιριάζει με την κανονική έκφραση μετά την κάθετο πχ 0/1 ταιριάζει με το 0 στην έκφραση 01 αλλά δεν ταιριάζει με κανένα χαρακτήρα στο αλφαριθμητικό 02

Περιεχόμενα Γραπτής Αναφοράς

- 1) Οι απαντήσεις στις ερωτήσεις που τίθενται