ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ · ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΑΡΧΕΣ ΓΛΩΣΣΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ & ΜΕΤΑΦΡΑΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2022

1α ΕΡΩΤΗΜΑ

```
<int> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<float> ::= <int> "." <int>

<letter> ::= "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" |
"l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" |
"y" | "z" | "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" |
"L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" |
"Y" | "Z"

<alphanumeric> ::= <letter> | <int>

<word> ::= <alphanumeric> | <alphanumeric> <word>

<json> ::= <object>

<object> ::= "{" "}" | "{" <content> "}"

<content> ::= <pair> | <pair> "," <content>

<pair> ::= <string> ":" <string> | <string> ":" <int> | <string> ":" <float> |
           <string> ":" <object> | <string> ":" <table>

<table> ::= "[" "]" | "[" <integers> "]" | "[" <objects> "]"

<objects> ::= <object> | <object> "," <objects>

<integers> ::= <int> | <int> "," <integers>

<string> ::= "\"" <keyword_tokens> "\""

<keyword_tokens> ::= "last" | "active" | "game_id" | "draw_id" | "draw_id" |
                     "draw_time" | "status" | "draw_break" | "visualDraw" |
                     "pricePoints" | "winningNumbers" | "prizeCategories" |
                     "wagerStatistics" | "amount" | "list" | "bonus" | "id" |
                     "divident" | "winners" | "distributed" | "categoryType" |
                     "gameType" | "minimumDistributed" | "columns" | "wagers" |
                     "addOn" | <word>
```

Καταρχήν δημιουργούμε τους τύπους των μεταβλητών που θα χρησιμοποιηθούν στην γλώσσα, δηλαδή `<float>`, `<int>` και `<letter>` τα οποία δύο τελευταία δημιουργούν και το αλφαριθμητικό `<alphanumeric>` και συνεπαγωγικά το `<word>`.

Η γλώσσα (`<json>`) ουσιαστικά αποτελείται από ένα `<object>`. Το `<object>` περιλαμβάνει δύο κανόνες που δημιουργούν το κενό object (`{}`) και το object με περιεχόμενο. Ορίζουμε το περιεχόμενο ως `<content>`. Το `<content>` περιλαμβάνει ζευγάρια (pairs) που αντιστοιχούν στη τυπολογία "`στοιχείο`": "`περιεχόμενο`" της JSON. Το content διαχωρίζει σε δύο κανόνες την πιθανότητα να περιλαμβάνει ένα μόνο ζευγάρι ή δύο και παραπάνω. Στην δεύτερη περίπτωση, προσθέτει ένα κόμμα ενδιάμεσα.

Όσον αφορά στο σχεδιασμό των ζευγαριών, πρόκειται για δύο στοιχεία που διαχωρίζονται με το σύμβολο ":". Το πρώτο στοιχείο πάντα θα είναι της μορφής "`αλφαριθμητικό`". Ορίζουμε ως `<string>` ένα αλφαριθμητικό που έχει prefix και suffix τα εισαγωγικά. Το δεύτερο στοιχείο μπορεί να είναι string, int, float, object ή table. Το `<string>` έχει διττό ρόλο καθώς όταν χρησιμοποιείται ως πρώτο στοιχείο περιλαμβάνει τις συγκεκριμένες λέξεις-κλειδιά (keywords) που αναπαριστούν το λεξιλόγιο της γλώσσας (τα οποία αποτελούν το `<keyword_tokens>`), αλλά ταυτόχρονα μπορεί να χρησιμοποιηθεί ως "`περιεχόμενο`", με άπειρα πιθανά αλφαριθμητικά. Για τη περίπτωση της απειρότητας, στο `<keyword_tokens>` περιλαμβάνεται και το `<word>`.

Τέλος ορίζουμε το table, που ή θα είναι κενό (`[]`), ή θα περιλαμβάνει ακεραίους (`[<integers>]`), ή θα περιλαμβάνει objects (`[<objects>]`). Τα `<integers>` είναι ή ένας ακέραιος ή πολλοί που διαχωρίζονται με κόμμα, και με παρόμοιο τρόπο ορίζονται και τα `<objects>`.

Η συγκεκριμένη BNF γραμματική περιγράφει λεξικολογικά τις προδιαγραφές της γλώσσας αλλά προφανώς δεν μπορεί να ελέγξει αν κάποια λέξη ευσταθεί στο λεξιλόγιο της γραμματικής, ούτε μπορεί να ελέγξει συντακτικά προδιαγραφές όπως το τι στοιχεία πρέπει να περιλαμβάνει ένα στοιχείο, ή το εύρος ενός ακεραίου.

1β ΕΡΩΤΗΜΑ

FLEX myJSONflexer.l

Ο κώδικας του Flex βρίσκεται στο τέλος του pdf.

Στο πρώτο section του αρχείου, στο C κομμάτι του, κάνουμε include τις βιβλιοθήκες που χρειάζονται, όπως επίσης και το header του Bison αρχείου ώστε να συνδεθεί με το Flex.

Στην συνέχεια ορίζουμε τα διαφορετικά λεξικολογικά στοιχεία με παρόμοιο τρόπο όπως με την BNF γραμματική. Συγκεκριμένα ορίζονται οι μορφολογίες των ακεραίων (`INT`), των πραγματικών (`FLOAT`) και των αλφαριθμητικών (`WORD`), τα οποία τελευταία ορίζονται με την βοήθεια των `LETTER` και `INT`. Επειδή μπορούμε να αξιοποιήσουμε τα regular expressions δεν είναι απαραίτητη η χρήση του `<alphanumeric>` για τον ορισμό της `WORD`[1]. Επίσης ορίζονται τα σημεία στίξης `QUOTE`, `COMMA`, `COLON`, `BRACKET_LEFT`, `BRACKET_RIGHT`, `BRACE_LEFT`, `BRACE_RIGHT` και το whitespace. Επίσης ορίζονται όλα τα keywords που χρησιμοποιούνται από τη γλώσσα, οι λέξεις δηλαδή που περιλαμβάνονταν στο `<keyword_tokens>` της BNF.

Οι επιλογές που ορίζονται με το prefix `%option` καθορίζουν το να αντιμετωπίζονται διαφορετικά οι κεφαλαίοι από τους πεζούς χαρακτήρες, να γίνεται αρίθμηση των γραμμών και το ότι ο parser δεν χρειάζεται να συνενώσει/ελέγξει πολλαπλά αρχεία, παρά μόνο ένα.

Στο δεύτερο section του αρχείου τίθενται οι κανόνες που θα ακολουθηθούν για την αναγνώριση των tokens. Επειδή ζητούμενο είναι η επιστροφή του αρχικού input, κάθε φορά που βρίσκεται ένα token, εκτυπώνεται κι όλας. Αυτό επιτυγχάνεται μέσω της μεταβλητής `yytext`, στην οποία αποθηκεύεται κάθε χρονική στιγμή το πιο πρόσφατο token που αναγνωρίζεται. Το αποτέλεσμα όλων αυτών των επί μέρους εκτυπώσεων είναι η **εμφάνιση του αρχικού input** στο command line. Τέλος, κάθε token επιστρέφεται επίσης ως `T_TOKEN`, ώστε να χρησιμοποιηθεί στο Bison για τους απαιτούμενους ελέγχους.

Παρατηρούμε ότι το `{WORD}` ορίζεται τελευταίο, σύμφωνα με την σειρά προτεραιότητας ορισμού κανόνων του Flex. Αν το ορίζαμε πάνω πάνω τότε θα "έκλεβε" την προτεραιότητα των υπόλοιπων keywords.

Η μεταβλητή `yylineno`, η οποία προέρχεται από το Flex/Bison και για αυτό ορίζεται ως external στην αρχή, την ορίζουμε να αυξάνεται κατά 1 κάθε φορά που αναγνωρίζεται το `\n`, συνεπώς αποτελεί το **μετρητή γραμμών** στην εμφάνιση σφαλμάτων όπως ζητείται στην εκφώνηση.

Δεν υπάρχει κομμάτι C στο τρίτο section του Flex, καθώς έχει μεταφερθεί όλο στο Bison.

[1] Αυτό συμβαίνει γιατί χρησιμοποιούμε επαναληπτικά operators όπως το "+".

BISON myJSONbison.y

Ο κώδικας του Bison βρίσκεται στο τέλος του pdf.

Στο τρίτο section του Bison, η main δέχεται τα κλασικά arguments της C που αφορούν το εκτελέσιμο πρόγραμμα που δημιουργείται, και τίθεται η δυνατότητα αυτόνομης εκτέλεσης του parser με το input του parser (`yyin`) να βρίσκεται στο command-line (αυτό συμβαίνει στην περίπτωση που `argc` = 0), είτε ο parser να διαβάζει ένα ξεχωριστό αρχείο, το filename του οποίου πρέπει να δοθεί ως argument μαζί με την εκτέλεση του προγράμματος.

Στη συνέχεια εκτελείται μια do-while επανάληψη στην οποία πρακτικά ξεκινάει να λειτουργεί ο parser έως ότου αναγνωρίσει EOF (end of file). Επίσης περιλαμβάνεται μια αρχικοποίηση μνήμης για τον string πίνακα `history` που θα χρησιμεύσει στη συνέχεια. Η `main` περιλαμβάνει και έναν επαναορισμό της συνάρτησης `yyerror` του Bison, στην οποία περιλαμβάνεται ο αριθμός της γραμμής[2] και η περιγραφή του σφάλματος.

Στο πρώτο section γίνονται include όλα τα απαραίτητα headers και ορίζονται όλες οι μεταβλητές (external ή internal), τα flags kai τα counters που θα χρησιμοποιηθούν. Επίσης καταγράφονται όλα τα διαφορετικά tokens όπως έχουν οριστεί στο Flex αρχείο ως `%token`. Σε αυτά περιλαμβάνεται ένα πρόθεμα που ορίζει το τύπο τους. Ο ορισμός των τύπων των μεταβλητών γίνεται μέσω του `%union`, το οποίο περιλαμβάνει όλες τις δυνατές μορφές (`intval`, `flval`, `strval`) που μπορεί να πάρει η μεταβλητή `yylval` στο πρόγραμμα. Επίσης ορίζονται και οι τύποι των nonterminal tokens μέσω του `%type <τύπος>`, τα οποία χρησιμοποιούνται στα rules. Έχει επίσης οριστεί και η προτεραιότητα των σημείων στίξης με το `%left`.

Ο τρόπος που έχουν συνταχτεί τα rules στο δεύτερο section έρχεται σε αντιστοιχία με την BNF γραμματική και το Flex. Το σημαντικό είναι το ότι τα μεταφέρουμε τις semantic τιμές των χρήσιμων nonterminal tokens στην στο αριστερό στοιχείο. Για παράδειγμα:

```
content : pair {$$ = $1;}
        | pair T_COMMA content {$$ = $1;} ;
```

Εδώ το περιεχόμενο του `pair` μεταφέρεται στο `content`, έτσι όταν ζητείται το `content` επιστρέφεται το `pair`, κάτι που είναι αναγκαίο για τους ελέγχους που θα πραγματοποιηθούν.

Με παρόμοιο τρόπο, αυτή η μεταφορά των semantic τιμών πραγματοποιείται σε όλα τα rules. Αν πρόκειται για αλφαριθμητικά όπως στα `keyword_tokens` χρησιμοποιείται η συνάρτηση `strdup` της βιβλιοθήκης `string.h` της C, όπως και παρόμοιες όπως η `strcmp` για τις εν δυνάμει συγκρίσεις[3].

Οι έλεγχοι πραγματοποιούνται κυρίως την ώρα που ο parser βρίσκεται στα pairs. Αυτό συμβαίνει γιατί υπάρχει άμεση πρόσβαση στα στοιχεία[4].

Γίνονται 3 είδη ελέγχου:

- Ελέγχεται ο τύπος των στοιχείων (για παράδειγμα το `gameId` πρέπει να είναι ακέραιος)
- Ελέγχεται το αν περιλαμβάνονται τα σωστά keywords και μόνο (το `not_gameId` δεν γίνει δεκτό σε οποιοδήποτε σημείο και αν βρίσκεται, όποιος και αν είναι ο τύπος του)
- Ελέγχεται αν τα στοιχεία βρίσκονται (μέσα) σε σωστό μέρος (το `list` αναγκαστικά πρέπει να περιέρχεται στο `winningNumbers`).

[2] Για κάποιο λόγο το yylineno μέτραγε διπλά τις μετρήσεις του \n, καταλήγοντας να εμφανίζει τον διπλάσιο αριθμό γραμμής από αυτό που θα έπρεπε κανονικά, εξού και το "yylineno/2+1". Εναλλακτικά βέβαια θα μπορούσα απλά να βρω γιατί διπλασιάζεται εξ αρχής.

[3] Είναι γνωστή η αναπηρία της C να χειρίζεται τα αλφαριθμητικά με τρόπο παρόμοιο όπως τους υπόλοιπους τύπους μεταβλητών.

[4] Αν ο έλεγχος γινόταν στο object πχ θα έπρεπε πρώτα να τυπωθεί το τελικό "}" πριν τυπωθούν τα τυχόντα σφάλματα. Στα pairs από την άλλη ελέγχονται και τυπώνονται άμεσα.

Έτσι όπως έχουν δομηθεί τα rules των pairs έχουμε σε κατηγορίες τους διαφορετικούς τύπους που μπορούν να πάρουν τα στοιχεία της γλώσσας:

```
pair    : string T_COLON string
        | string T_COLON T_INT
        | string T_COLON T_FLOAT
        | string T_COLON object
        | string T_COLON table
```

Άρα για να βεβαιωθώ ότι ένα στοιχείο είναι στο τύπο που θέλω, πρέπει όταν ο parser το ανιχνεύσει και πάει στην αντίστοιχη υποπερίπτωση ενός εκ των από πάνω κανόνων, να βρίσκεται *μόνο* εκεί και όχι στους υπόλοιπους κανόνες. Μέχρι στιγμής αν όριζα στο input `"gameId" = "string"`, θα ήταν αποδεκτό και θα οδηγούμασταν στον πρώτο κανόνα. Συνεπώς πρέπει να ορίσουμε **κάθε στοιχείο να μην υπάρχει σε όλους τους υπόλοιπους κανόνες παρά μόνο στο σωστό**. Αυτή είναι η δουλειά του τμήματος `/* Έλεγχος τύπου στοιχείων */` του κώδικα. `$1` είναι το πρώτο string, δηλαδή κάθε keyword (για αυτό το λόγο είναι απαραίτητη η μεταφορά της semantic τιμής), και συγκρίνεται με όλες τις μη-δυνατές τιμές που μπορεί να έχει από τη λίστα των keywords. Αν πράγματι δεν θα έπρεπε να ανήκει εκεί, καλείται η τροποποιημένη `yyerror` αναφέροντας την γραμμή του σφάλματος.

Αυτός ο έλεγχος βρίσκεται σε καθένα από τα 5 rules περιλαμβάνοντας όλα τα συμπληρωματικά στοιχεία σε καθένα και αφορά την λανθασμένη χρήση των υπαρχόντων keywords. Ειδικές περιπτώσεις στον έλεγχο των τύπων των υπαρκτών keywords αποτελούν τα `id` και `categoryType` που αφορούν ακεραίους **με συγκεκριμένο εύρος**. Σε αυτές τις περιπτώσεις ελέγχεται **και** το `$3`, δηλαδή ο ίδιος ο ακέραιος για το αν ανήκει στις σωστές συνθήκες.

Αν το $1 τύχει να είναι κάποιο στοιχείο που δεν είναι καν keyword οδηγούμαστε στο `/* Έλεγχος άκυρων στοιχείων */` το οποίο ελέγχει αν το `$1` είναι κάποιο από τα στοιχεία που πράγματι πρέπει να είναι (δηλαδή τα συμπληρωματικά των συμπληρωματικών). Αν δεν είναι, τυπώνεται και πάλι το αντίστοιχο σφάλμα.

Μέχρι στιγμής δεν έχει καλυφθεί η περίπτωση του να είναι σωστά ορισμένο, σε λάθος μέρος (για παράδειγμα το `gameId` να βρίσκεται μέσα στο `winningNumbers` αντί για το `last` ή το `active`). Εδώ χρησιμεύει το string array `history` των οποίων οι θέσεις μνήμης αρχικοποιήθηκαν στην `main`.

Στον κανόνα `string` που προσθέτει στα tokens τα εισαγωγικά, δημιουργείται ένας στατικός μετρητής `i` όπου παρέα με την `history` έχουν αποθηκεύσει και καταμετρήσει κάθε token που γίνεται parse από την είσοδο. Αυτό είναι ένα παράδειγμα από το αποτέλεσμα που έχει δημιουργηθεί:

```
    "last"(i=1): {
        "gameId"(i=2): 5104,
        "drawId"(i=3): 2390,
        "drawTime"(i=4): 1642363200000,
        "status"(i=5): "results"(i=6),
        "drawBreak"(i=7): 1800000
                                ... ... ...
        "wagerStatistics"(i=176): {
            "wagers"(i=178): 0,
            "addOn"(i=179): []
```

(το συγκεκριμένο αρχείο με τα i περιλαμβάνεται στο .zip)

Άρα πλέον κάθε token έχει **συγκεκριμένη τοποθεσία** με αρνητικό το ότι δεν μπορεί να γίνει κάποια εσωτερική μετακίνηση μέσα στο ίδιο object χωρίς να επηρεαστούν τα δεδομένα.[5] Έτσι μπορούμε να

[5] Ένας τρόπος που μπορεί να λυθεί αυτό το ζήτημα είναι να τεθούν συγκεκριμένα **όρια** μετακίνησης των tokens. Το gameId πχ δεν μπορεί να έχει (i <2) ή (i > 11) και αντίστοιχα για όλα τα υπόλοιπα.

συγκρίνουμε το $1 με τα σωστά tokens που παίρνει σε κάθε rule του pair, και αν το `i` του (που συνεχώς αυξάνεται κατά την ανάγνωση του parser) δεν είναι σωστό, τυπώνεται το αντίστοιχο σφάλμα.Έτσι επιτυγχάνεται και η εμφάνιση του `minimumDistributed` μόνο όταν `"id":1`.

Μπορεί να χρησιμοποιηθεί η ίδια τεχνική και για την σειρά του last & active, παρόλα αυτά έχω χρησιμοποιήσει δύο flags, τα `last_was_found` και `active_was_found` που ενεργοποιούνται όταν αναγνωρίζεται κάποιο από αυτά τα tokens και με τις κατάλληλες συνθήκες περιορίζονται οι περιπτώσεις στο να αναγνωρίζεται πρώτα το `last` και μετά το `active`, αλλιώς τυπώνονται μηνύματα σφάλματος.

2 ΕΡΩΤΗΜΑ

Δεν έχει πραγματοποιηθεί.

3 ΕΡΩΤΗΜΑ

Το να περιλαμβάνει το `prizeCategories` 8 αντικείμενα έχει ήδη κλειδωθεί από την χρήση των `i`, κάτι το οποίο εναλλακτικά μπορεί να επιτευχθεί και με ένα flag και counter.

Η μόνη φορά που χρησιμοποιείται το `integers` είναι για την `list` του `winningNumbers`. Εκτελείται ο δεύτερος κανόνας για όλους τους ακεραίους πέρα από τον τελευταίο ακέραιο που σε αυτή τη περίπτωση χρησιμοποιείται ο πρώτος κανόνας. Ο πρώτος κανόνας χρησιμοποιείται σε κάθε περίπτωση, ακόμα και αν είχαμε μόνο έναν ακέραιο. Άρα ο δεύτερος κανόνας πρέπει να εκτελεστεί 5-1 = 4 φορές. Έτσι απλά δημιουργούμε τον κατάλληλο static counter. Αν δεν είναι 4, τότε εμφανίζει σφάλμα.

Όσον αφορά για το εύρος των ακεραίων, στο `T_INT T_COMMA integers` ή στο `T_INT` γίνεται ο έλεγχος στο εύρος του `$1` που αφορά τον ακέραιο.

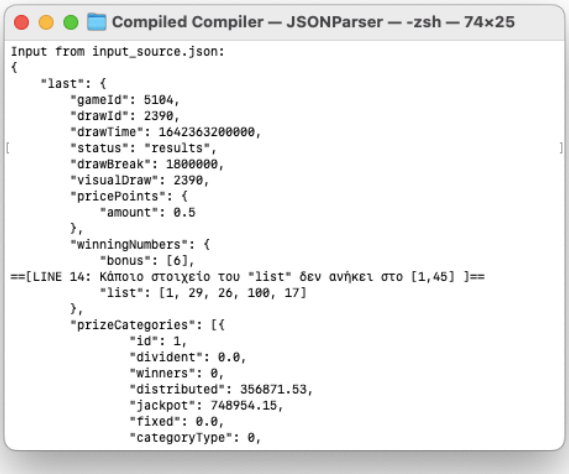
EXE JSONParser

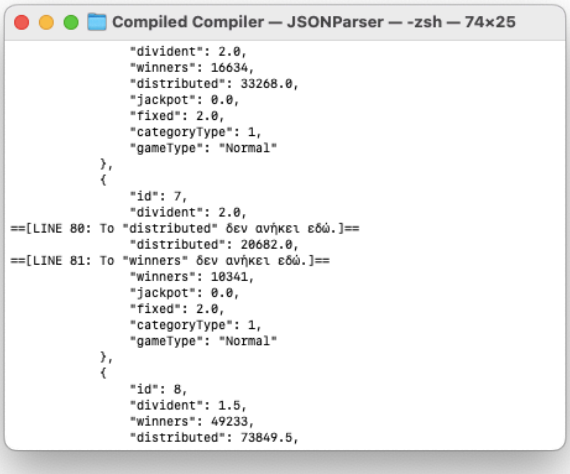
Στα παραδείγματα που θα ακολουθήσουν έχουν χρησιμοποιηθεί ως pipeline διαδοχικά τα εξής commands για το compiling και την εκτέλεση του parser:

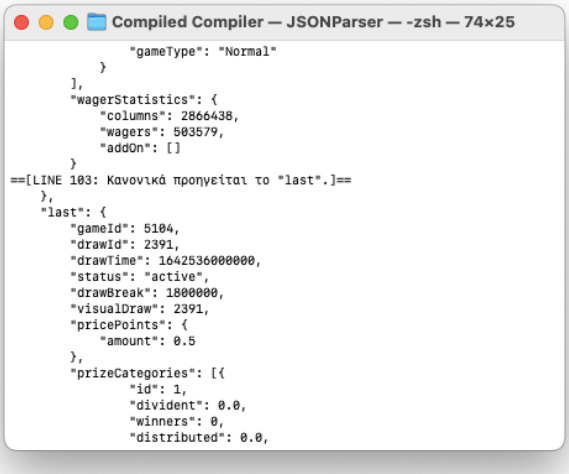
```
flex myJSONflexer.l;
bison myJSONbison.y;
gcc lex.yy.c -o JSONParser;
./JSONParser input_source.json
```

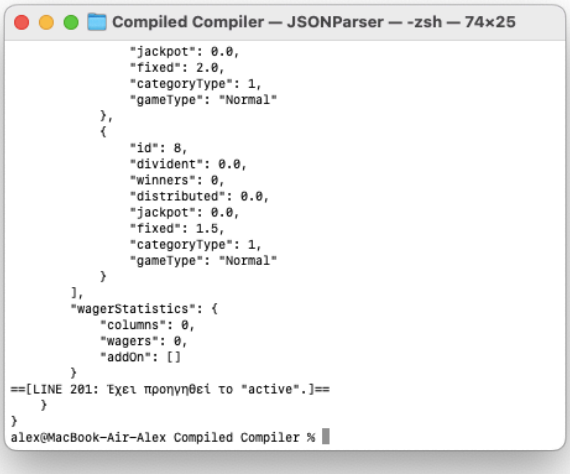
(το input file input_source.json είναι μετονομασμένο το last_result.json)

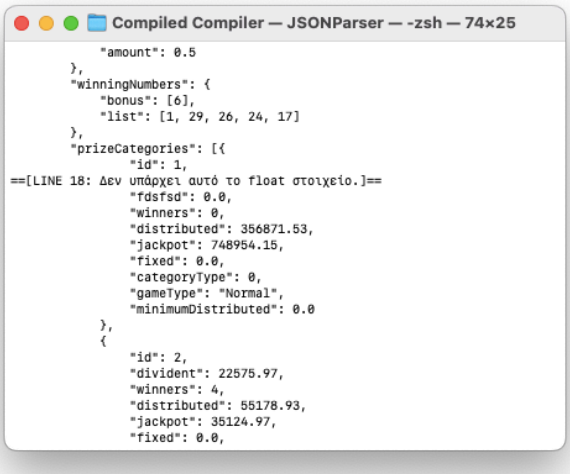
ΠΑΡΑΔΕΙΓΜΑΤΑ - SCREENSHOTS

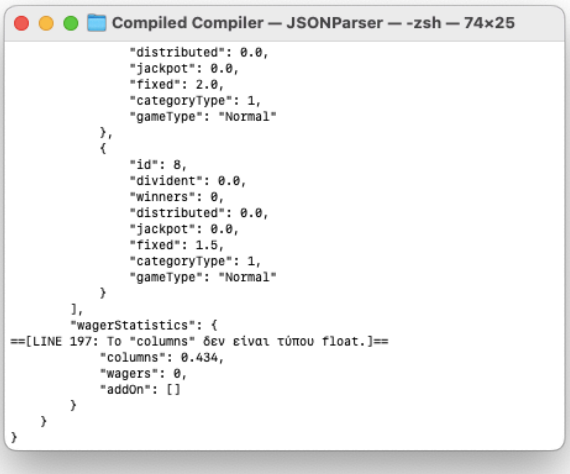












FLEX myJSONflexer.l

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
    #include <unistd.h>
    #include "myJSONbison.tab.c" //σύνδεση bison
    extern int yylineno;
%}
%option case-sensitive
%option yylineno
%option noyywrap

INT                 ([0-9])+
FLOAT               {INT}+\.{INT}+
LETTER              [a-zA-Z]
WORD                ({LETTER}|{INT})+

QUOTE               ["]
COMMA               ","
COLON               ":"
BRACKET_LEFT        "["
BRACKET_RIGHT       "]"
BRACE_LEFT          "{"
BRACE_RIGHT         "}"
SPACE               [ \t]
LAST                "last"
GAME_ID             "gameId"
DRAW_ID             "drawId"
DRAW_TIME           "drawTime"
STATUS              "status"
DRAW_BREAK          "drawBreak"
VISUAL_DRAW         "visualDraw"
PRICE_POINTS        "pricePoints"
AMOUNT              "amount"
WINNING_NUMBERS     "winningNumbers"
LIST                "list"
BONUS               "bonus"
PRIZE_CATEGORIES    "prizeCategories"
ID                  "id"
DIVIDENT            "divident"
WINNERS             "winners"
DISTRIBUTED         "distributed"
JACKPOT             "jackpot"
FIXED               "fixed"
```

```
CATEGORY_TYPE       "categoryType"
GAME_TYPE           "gameType"
MINIMUM_DISTRIBUTED "minimumDistributed"
WAGER_STATISTICS    "wagerStatistics"
COLUMNS             "columns"
WAGERS              "wagers"
ADD_ON              "addOn"
ACTIVE              "active"
%%
{QUOTE}                     {printf("%s", yytext); return T_QUOTE;}
{COLON}                     {printf("%s", yytext); return T_COLON;}
{COMMA}                     {printf("%s", yytext); return T_COMMA;}
{BRACKET_LEFT}              {printf("%s", yytext); return T_LBRACKET;}
{BRACKET_RIGHT}             {printf("%s", yytext); return T_RBRACKET;}
{BRACE_LEFT}                {printf("%s", yytext); return T_LBRACE;}
{BRACE_RIGHT}               {printf("%s", yytext); return T_RBRACE;}

{INT}                    {printf("%s", yytext); yylval.intval = atoi(yytext);
                                                 return T_INT;}
{FLOAT}                   {printf("%s", yytext); yylval.flval = atof(yytext);
                                                 return T_FLOAT;}

{LAST}                  {printf("%s", yytext); yylval.strval = strdup(yytext);
return T_LAST;}
{ACTIVE}                 {printf("%s", yytext); return T_ACTIVE;}
{GAME_ID}                {printf("%s", yytext); return T_GAME_ID;}
{DRAW_ID}                {printf("%s", yytext); return T_DRAW_ID;}
{DRAW_TIME}              {printf("%s", yytext); return T_DRAW_TIME;}
{STATUS}                 {printf("%s", yytext); return T_STATUS;}
{DRAW_BREAK}             {printf("%s", yytext); return T_DRAW_BREAK;}
{VISUAL_DRAW}            {printf("%s", yytext); return T_VISUAL_DRAW;}
{PRICE_POINTS}           {printf("%s", yytext); return T_PRICE_POINTS;}
    {AMOUNT}             {printf("%s", yytext); return T_AMOUNT;}
{WINNING_NUMBERS}        {printf("%s", yytext); return T_WINNING_NUMBERS;}
    {LIST}               {printf("%s", yytext); return T_LIST;}
    {BONUS}              {printf("%s", yytext); return T_BONUS;}
{PRIZE_CATEGORIES}       {printf("%s", yytext); return T_PRIZE_CATEGORIES;}
    {ID}                 {printf("%s", yytext); return T_ID;}
    {DIVIDENT}           {printf("%s", yytext); return T_DIVIDENT;}
    {WINNERS}            {printf("%s", yytext); return T_WINNERS;}
    {DISTRIBUTED}        {printf("%s", yytext); return T_DISTRIBUTED;}
    {JACKPOT}            {printf("%s", yytext); return T_JACKPOT;}
    {FIXED}              {printf("%s", yytext); return T_FIXED;}
    {CATEGORY_TYPE}      {printf("%s", yytext); return T_CATEGORY_TYPE;}
    {GAME_TYPE}          {printf("%s", yytext); return T_GAME_TYPE;}
    {MINIMUM_DISTRIBUTED} {printf("%s", yytext); return T_MINIMUM_DISTRIBUTED;}
```

```
{WAGER_STATISTICS}        {printf("%s", yytext); return T_WAGER_STATISTICS;}
    {COLUMNS}             {printf("%s", yytext); return T_COLUMNS;}
    {WAGERS}              {printf("%s", yytext); return T_WAGERS;}
    {ADD_ON}              {printf("%s", yytext); return T_ADD_ON;}
{WORD}                    {printf("%s", yytext); return T_WORD;}

\n                       {printf("\n"); yylineno++;}
{SPACE}                    {printf("%s", yytext);}
.                          {printf("");}
%%
```

BISON myJSONbison.y

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
    #define HISTORY_SIZE 200

    // Lex/Bison external variables
    extern int yylineno;
    extern int yylex();
    extern int yyparce();
    extern void yyerror(char *);
    extern char* yytext;
    extern FILE *yyin;
    extern FILE *yyout;

    // Flags
    static int last_was_found = 0;
    static int active_was_found = 0;
    int not_in_history = 0;

    // Counters
    int j = 0;
    static int i = 0;
    static int list_counter = 0;

    // Strings
    char* history[HISTORY_SIZE];
    char* temp_val = "";
%}

%union {
    char*   strval;
```

```
    int     intval;
    float   flval;
}

%token <intval> T_INT
%token <flval>  T_FLOAT
%token <strval> T_QUOTE T_COLON T_COMMA T_LBRACKET T_RBRACKET T_LBRACE T_RBRACE
%token <strval> T_WORD T_LAST T_ACTIVE T_GAME_ID T_DRAW_ID T_DRAW_TIME T_STATUS
T_DRAW_BREAK
%token <strval> T_VISUAL_DRAW T_PRICE_POINTS T_AMOUNT T_WINNING_NUMBERS T_LIST
T_BONUS T_PRIZE_CATEGORIES
%token <strval> T_ID T_DIVIDENT T_WINNERS T_DISTRIBUTED T_JACKPOT T_FIXED
T_CATEGORY_TYPE T_GAME_TYPE
%token <strval> T_MINIMUM_DISTRIBUTED T_WAGER_STATISTICS T_COLUMNS T_WAGERS
T_ADD_ON

%type <strval> string keyword_tokens object content pair table objects
%type <intval> integers

%left T_QUOTE T_COLON T_COMMA T_LBRACKET T_RBRACKET T_LBRACE T_RBRACE

%%
/* ========[ RULES ]======== */

json: object;

object  : T_LBRACE T_RBRACE
        | T_LBRACE content T_RBRACE {$$ = $2;}
        ;

content : pair {$$ = $1;}
        | pair T_COMMA content {$$ = $1;}
        ;

pair    : string T_COLON string { $$ = $1;

            /* Ελεγχος άκυρων στοιχείων */
            if (!((strcmp($1, "status") == 0) || (strcmp($1, "gameType") == 0)))
{
                yyerror("Δεν υπάρχει αυτό το στοιχείο.");
            }

            /* Ελεγχος τοποθέτησης στοιχείων στο σωστό μέρος */
            // Το i είναι +1 λόγω του δεύτερου string
            if (strcmp($1, "status") == 0) {
                if (i != 6 && i != 97)
                    yyerror("Το \"status\" δεν ανήκει εδώ.");
```

```
            }
            else if (strcmp($1, "gameType") == 0) {
                if (i != 23 && i != 33 && i != 42 && i != 51 && i != 60 && i !=
69 && i != 78 && i != 87 && i != 111
                    && i != 121 && i != 130 && i != 139 && i != 128 && i != 148
&& i != 157 && i != 166 && i != 175)
                    yyerror("Το \"gameType\" δεν ανήκει εδώ.");
            }

            /* Έλεγχος τύπου στοιχείων */
            if (strcmp($1, "gameId") == 0)
                yyerror("Το \"gameId\" δεν είναι τύπου string");
            else if (strcmp($1, "drawId") == 0)
                yyerror("Το \"drawId\" δεν είναι τύπου string");
            else if (strcmp($1, "drawTime") == 0)
                yyerror("Το \"drawTime\" δεν είναι τύπου string");
            else if (strcmp($1, "drawBreak") == 0)
                yyerror("Το \"drawBreak\" δεν είναι τύπου string");
            else if (strcmp($1, "visualDraw") == 0)
                yyerror("Το \"visualDraw\" δεν είναι τύπου string");
            else if (strcmp($1, "pricePoints") == 0)
                yyerror("Το \"pricePoints\" δεν είναι τύπου string");
            else if (strcmp($1, "winningNumbers") == 0)
                yyerror("Το \"winningNumbers\" δεν είναι τύπου string");
            else if (strcmp($1, "winningNumbers") == 0)
                yyerror("Το \"winningNumbers\" δεν είναι τύπου string");
            else if (strcmp($1, "wagerStatistics") == 0)
                yyerror("Το \"wagerStatistics\" δεν είναι τύπου string");
            else if (strcmp($1, "list") == 0)
                yyerror("Το \"list\" δεν είναι τύπου string.");
            else if (strcmp($1, "bonus") == 0)
                yyerror("Το \"bonus\" δεν είναι τύπου string.");
            else if (strcmp($1, "amount") == 0)
                yyerror("Το \"amount\" δεν είναι τύπου string.");
            else if (strcmp($1, "id") == 0)
                yyerror("Το \"id\" δεν είναι τύπου string.");
            else if (strcmp($1, "divident") == 0)
                yyerror("Το \"divident\" δεν είναι τύπου string.");
            else if (strcmp($1, "winners") == 0)
                yyerror("Το \"winners\" δεν είναι τύπου string.");
            else if (strcmp($1, "distributed") == 0)
                yyerror("Το \"distributed\" δεν είναι τύπου string.");
            else if (strcmp($1, "jackpot") == 0)
                yyerror("Το \"jackpot\" δεν είναι τύπου string.");
            else if (strcmp($1, "fixed") == 0)
                yyerror("Το \"fixed\" δεν είναι τύπου string.");
```

```
            else if (strcmp($1, "categoryType") == 0)
                yyerror("Το \"categoryType\" δεν είναι τύπου string.");
            else if (strcmp($1, "minimumDistributed") == 0)
                yyerror("Το \"minimumDistributed\" δεν είναι τύπου string.");
            else if (strcmp($1, "columns") == 0)
                yyerror("Το \"columns\" δεν είναι τύπου string.");
            else if (strcmp($1, "wagers") == 0)
                yyerror("Το \"wagers\" δεν είναι τύπου string.");
            else if (strcmp($1, "addOn") == 0)
                yyerror("Το \"addOn\" δεν είναι τύπου string.");

        }

        | string T_COLON T_INT { $$ = $1;

            /* Έλεγχος τοποθέτησης στοιχείων στο σωστό μέρος */
            if (strcmp($1, "gameId") == 0) {
                if (i != 2 && i != 93)
                    yyerror("Το \"gameId\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "drawId") == 0) {
                if (i != 3 && i != 94)
                        yyerror("Το \"drawId\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "drawTime") == 0) {
                if (i != 4 && i != 95)
                        yyerror("Το \"drawTime\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "drawBreak") == 0) {
                if (i != 7 && i != 98)
                    yyerror("Το \"drawBreak\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "visualDraw") == 0) {
                if (i != 8 && i != 99)
                        yyerror("Το \"visualDraw\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "id") == 0) {
                if (i != 15 && i != 25 && i != 34 && i != 43 && i != 52 && i !=
61 && i != 70 && i != 79 && i != 103
                    && i != 113 && i != 122 && i != 131 && i != 140 && i != 149
&& i != 158 && i != 167 && i != 175)
                        yyerror("Το \"id\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "winners") == 0) {
                if (i != 17 && i != 27 && i != 36 && i != 45 && i != 54 && i !=
63 && i != 72 && i != 81 && i != 105
```

```
                    && i != 115 && i != 124 && i != 133 && i != 142 && i != 151
&& i != 160 && i != 169)
                        yyerror("Το \"winners\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "columns") == 0) {
                if (i != 89 && i != 177)
                        yyerror("Το \"columns\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "wagers") == 0) {
                if (i != 90 && i != 178)
                        yyerror("Το \"wagers\" δεν ανήκει εδώ.");
            }

            /* Έλεγχος τύπου στοιχείων */
            if (strcmp($1, "status") == 0)
                yyerror("Το \"status\" δεν είναι τύπου int");
            else if (strcmp($1, "pricePoints") == 0)
                yyerror("Το \"pricePoints\" δεν είναι τύπου int");
            else if (strcmp($1, "winningNumbers") == 0)
                yyerror("Το \"winningNumbers\" δεν είναι τύπου int");
            else if (strcmp($1, "winningNumbers") == 0)
                yyerror("Το \"winningNumbers\" δεν είναι τύπου int");
            else if (strcmp($1, "wagerStatistics") == 0)
                yyerror("Το \"wagerStatistics\" δεν είναι τύπου int");
            else if (strcmp($1, "list") == 0)
                yyerror("Το \"list\" δεν είναι τύπου int.");
            else if (strcmp($1, "bonus") == 0)
                yyerror("Το \"bonus\" δεν είναι τύπου int.");
            else if (strcmp($1, "amount") == 0)
                yyerror("Το \"amount\" δεν είναι τύπου int.");
            else if (strcmp($1, "id") == 0) {
                if (($3 > 8) || ($3 < 1))   //έλεγχος εύρος id
                    yyerror("Το \"id\" δεν περιλαμβάνει σωστό εύρος ακεραίων.");
            }
            else if (strcmp($1, "divident") == 0)
                yyerror("Το \"divident\" δεν είναι τύπου int.");
            else if (strcmp($1, "distributed") == 0)
                yyerror("Το \"distributed\" δεν είναι τύπου int.");
            else if (strcmp($1, "jackpot") == 0)
                yyerror("Το \"jackpot\" δεν είναι τύπου int.");
            else if (strcmp($1, "fixed") == 0)
                yyerror("Το \"fixed\" δεν είναι τύπου int.");
            else if (strcmp($1, "categoryType") == 0) {
                if (($3 > 2) || ($3 < 0))   //έλεγχος εύρος categoryType
                    yyerror("Το \"categoryType\" δεν περιλαμβάνει σωστό εύρος
ακεραίων.");
```

```
            }
            else if (strcmp($1, "gameType") == 0)
                yyerror("Το \"gameType\" δεν είναι τύπου int.");
            else if (strcmp($1, "minimumDistributed") == 0)
                yyerror("Το \"minimumDistributed\" δεν είναι τύπου int.");
            else if (strcmp($1, "addOn") == 0)
                yyerror("Το \"addOn\" δεν είναι τύπου int.");

            /* Έλεγχος άκυρων στοιχείων */
            else if (!((strcmp($1, "gameId") == 0) || (strcmp($1, "drawId") ==
0) || (strcmp($1, "drawTime") == 0) || (strcmp($1, "drawBreak") == 0) ||
(strcmp($1, "fixed") == 0) || (strcmp($1, "visualDraw") == 0)
                    || (strcmp($1, "id") == 0) || (strcmp($1, "winners") == 0)
|| (strcmp($1, "categoryType") == 0) || (strcmp($1, "columns") == 0) ||
(strcmp($1, "wagers") == 0) ))
                yyerror("Δεν υπάρχει αυτό το int στοιχείο.");
        }

        | string T_COLON T_FLOAT {
            /* Έλεγχος τοποθέτησης στοιχείων στο σωστό μέρος */
            if (strcmp($1, "amount") == 0) {
                if (i != 10 && i != 101)
                    yyerror("Το \"amount\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "divident") == 0) {
                if (i != 16 && i != 26 && i != 35 && i != 44 && i != 53 && i !=
62 && i != 71 && i != 80 && i != 104
                    && i != 114 && i != 123 && i != 132 && i != 141 && i != 150
&& i != 159 && i != 168)
                    yyerror("Το \"divident\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "distributed") == 0) {
                if (i != 18 && i != 28 && i != 37 && i != 46 && i != 55 && i !=
64 && i != 73 && i != 82 && i != 106
                    && i != 116 && i != 125 && i != 134 && i != 143 && i != 152
&& i != 161 && i != 170)
                    yyerror("Το \"distributed\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "jackpot") == 0) {
                if (i != 19 && i != 29 && i != 38 && i != 47 && i != 56 && i !=
65 && i != 74 && i != 83 && i != 107
                    && i != 117 && i != 126 && i != 135 && i != 144 && i != 153
&& i != 162 && i != 171)
                    yyerror("Το \"jackpot\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "fixed") == 0) {
                if (i != 20 && i != 30 && i != 39 && i != 48 && i != 57 && i !=
66 && i != 75 && i != 84 && i != 108
```

```
                    && i != 118 && i != 127 && i != 136 && i != 145 && i != 154
&& i != 163 && i != 172)
                    yyerror("Το \"fixed\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "minimumDistributed") == 0) {
                if (i != 24 && i != 112)
                    yyerror("Το \"amount\" δεν ανήκει εδώ.");
            }

            /* Έλεγχος τύπου δεδομένων */
            if (strcmp($1, "status") == 0)
                yyerror("Το \"status\" δεν είναι τύπου float");
            else if (strcmp($1, "gameId") == 0)
                yyerror("Το \"gameId\" δεν είναι τύπου float");
            else if (strcmp($1, "drawId") == 0)
                yyerror("Το \"drawId\" δεν είναι τύπου float");
            else if (strcmp($1, "drawTime") == 0)
                yyerror("Το \"drawTime\" δεν είναι τύπου float");
            else if (strcmp($1, "drawBreak") == 0)
                yyerror("Το \"drawBreak\" δεν είναι τύπου float");
            else if (strcmp($1, "visualDraw") == 0)
                yyerror("Το \"visualDraw\" δεν είναι τύπου float");
            else if (strcmp($1, "pricePoints") == 0)
                yyerror("Το \"pricePoints\" δεν είναι τύπου float.");
            else if (strcmp($1, "winningNumbers") == 0)
                yyerror("Το \"winningNumbers\" δεν είναι τύπου float.");
            else if (strcmp($1, "prizeCategories") == 0)
                yyerror("Το \"prizeCategories\" δεν είναι τύπου float.");
            else if (strcmp($1, "wagerStatistics") == 0)
                yyerror("Το \"wagerStatistics\" δεν είναι τύπου float.");
            else if (strcmp($1, "list") == 0)
                yyerror("Το \"list\" δεν είναι τύπου float.");
            else if (strcmp($1, "bonus") == 0)
                yyerror("Το \"bonus\" δεν είναι τύπου float.");
            else if (strcmp($1, "id") == 0)
                yyerror("Το \"id\" δεν είναι τύπου float.");
            else if (strcmp($1, "winners") == 0)
                yyerror("Το \"winners\" δεν είναι τύπου float.");
            else if (strcmp($1, "categoryType") == 0)
                yyerror("Το \"categoryType\" δεν είναι τύπου float.");
            else if (strcmp($1, "gameType") == 0)
                yyerror("Το \"gameType\" δεν είναι τύπου float.");
            else if (strcmp($1, "columns") == 0)
                yyerror("Το \"columns\" δεν είναι τύπου float.");
            else if (strcmp($1, "wagers") == 0)
```

```
                yyerror("Το \"wagers\" δεν είναι τύπου float.");
            else if (strcmp($1, "addOn") == 0)
                yyerror("Το \"addOn\" δεν είναι τύπου float.");

            /* Ελεγχος άκυρων στοιχείων */
            else if (!((strcmp($1, "amount") == 0) || (strcmp($1, "divident")
== 0) || (strcmp($1, "distributed") == 0) || (strcmp($1, "jackpot") == 0) ||
(strcmp($1, "fixed") == 0) || (strcmp($1, "minimumDistributed") == 0)))
                yyerror("Δεν υπάρχει αυτό το float στοιχείο.");
        }

        | string T_COLON object { $$ = $1;
            /* Ελεγχος προτεραιότητας last - active */
            if (strcmp($1, "last") == 0) {
                if (active_was_found == 1 && last_was_found == 0)
                    yyerror("Εχει προηγηθεί το \"active\".");
                last_was_found = 1;
            }
            if (strcmp($1, "active") == 0) {
                if (active_was_found == 1)
                    yyerror("Εχει προηγηθεί το \"active\".");
                else if (last_was_found == 0 && active_was_found == 0)
                    yyerror("Κανονικά προηγείται το \"last\".");
                active_was_found = 1;
            }

            /* Ελεγχος τοποθέτησης στοιχείων στο σωστό μέρος */
            //Το i είναι το τελευταίο στοιχείο του { }
            if (strcmp($1, "pricePoints") == 0) {
                if (i != 10 && i != 101)
                    yyerror("Το \"pricePoints\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "winningNumbers") == 0) {
                if (i != 13)
                    yyerror("Το \"winningNumbers\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "wagerStatistics") == 0) {
                if (i != 91 && i != 179)
                    yyerror("Το \"wagerStatistics\" δεν ανήκει εδώ.");
            }

            /* Ελεγχος περιεχομένων pricePoints */
            if (strcmp($1, "pricePoints") == 0)
                if (strcmp($3, "amount") != 0)
                    yyerror("Το \"pricePoints\" περιλαμβάνει μόνο το amount.");
```

```
            /* Έλεγχος περιεχομένων winningNumbers */
            // if (strcmp($1, "winningNumbers") == 0) {
            //     if ( strcmp(pair, "list") != 0 || (strcmp(content, "bonus")
!= 0) )
            //         yyerror("Το winningNumbers περιλαμβάνει μόνο το list και
το bonus.");
            // }

            /* Έλεγχος τύπου δεδομένων */
            if (strcmp($1, "status") == 0)
                yyerror("Το \"status\" δεν είναι τύπου object");
            else if (strcmp($1, "gameId") == 0)
                yyerror("Το \"gameId\" δεν είναι τύπου object");
            else if (strcmp($1, "drawId") == 0)
                yyerror("Το \"drawId\" δεν είναι τύπου object");
            else if (strcmp($1, "drawTime") == 0)
                yyerror("Το \"drawTime\" δεν είναι τύπου object");
            else if (strcmp($1, "drawBreak") == 0)
                yyerror("Το \"drawBreak\" δεν είναι τύπου object");
            else if (strcmp($1, "visualDraw") == 0)
                yyerror("Το \"visualDraw\" δεν είναι τύπου object");
            else if (strcmp($1, "list") == 0)
                yyerror("Το \"list\" δεν είναι τύπου object.");
            else if (strcmp($1, "bonus") == 0)
                yyerror("Το \"bonus\" δεν είναι τύπου object.");
            else if (strcmp($1, "amount") == 0)
                yyerror("Το \"amount\" δεν είναι τύπου object.");
            else if (strcmp($1, "id") == 0)
                yyerror("Το \"id\" δεν είναι τύπου object.");
            else if (strcmp($1, "divident") == 0)
                yyerror("Το \"divident\" δεν είναι τύπου object.");
            else if (strcmp($1, "winners") == 0)
                yyerror("Το \"winners\" δεν είναι τύπου object.");
            else if (strcmp($1, "distributed") == 0)
                yyerror("Το \"distributed\" δεν είναι τύπου object.");
            else if (strcmp($1, "jackpot") == 0)
                yyerror("Το \"jackpot\" δεν είναι τύπου object.");
            else if (strcmp($1, "fixed") == 0)
                yyerror("Το \"fixed\" δεν είναι τύπου object.");
            else if (strcmp($1, "categoryType") == 0)
                yyerror("Το \"categoryType\" δεν είναι τύπου object.");
            else if (strcmp($1, "gameType") == 0)
                yyerror("Το \"gameType\" δεν είναι τύπου object.");
            else if (strcmp($1, "minimumDistributed") == 0)
                yyerror("Το \"minimumDistributed\" δεν είναι τύπου object.");
            else if (strcmp($1, "columns") == 0)
```

```
                yyerror("Το \"columns\" δεν είναι τύπου object.");
            else if (strcmp($1, "wagers") == 0)
                yyerror("Το \"wagers\" δεν είναι τύπου object.");
            else if (strcmp($1, "addOn") == 0)
                yyerror("Το \"addOn\" δεν είναι τύπου object.");

            /* Έλεγχος άκυρων στοιχείων */
            else if (!((strcmp($1, "last") == 0) || (strcmp($1,
"winningNumbers") == 0) || (strcmp($1, "wagerStatistics") == 0)
                            || (strcmp($1, "active") == 0) || (strcmp($1,
"pricePoints") == 0))) {
                yyerror("Δεν υπάρχει αυτό το object στοιχείο.");
            }

        }
        | string T_COLON table { $$ = $1;
            /* Έλεγχος τοποθέτησης στοιχείων στο σωστό μέρος */
            if (strcmp($1, "bonus") == 0) {
                if (i != 12)
                    yyerror("Το \"bonus\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "list") == 0) {
                if (i != 13)
                    yyerror("Το \"list\" δεν ανήκει εδώ.");
            }
            else if (strcmp($1, "addOn") == 0) {
                if (i != 91 && i != 179)
                    yyerror("Το \"addOn\" δεν ανήκει εδώ.");
            }

            /* Έλεγχος τύπου στοιχείων */
            if (strcmp($1, "status") == 0)
                yyerror("Το \"status\" δεν είναι τύπου table");
            else if (strcmp($1, "gameId") == 0)
                yyerror("Το \"gameId\" δεν είναι τύπου table");
            else if (strcmp($1, "drawId") == 0)
                yyerror("Το \"drawId\" δεν είναι τύπου table");
            else if (strcmp($1, "drawTime") == 0)
                yyerror("Το \"drawTime\" δεν είναι τύπου table");
            else if (strcmp($1, "drawBreak") == 0)
                yyerror("Το \"drawBreak\" δεν είναι τύπου table");
            else if (strcmp($1, "visualDraw") == 0)
                yyerror("Το \"visualDraw\" δεν είναι τύπου table");
            else if (strcmp($1, "pricePoints") == 0)
                yyerror("Το \"pricePoints\" δεν είναι τύπου table.");
            else if (strcmp($1, "winningNumbers") == 0)
```

```
                yyerror("To \"winningNumbers\" δεν είναι τύπου table.");
            else if (strcmp($1, "wagerStatistics") == 0)
                yyerror("To \"wagerStatistics\" δεν είναι τύπου table.");
            else if (strcmp($1, "amount") == 0)
                yyerror("To \"amount\" δεν είναι τύπου table.");
            else if (strcmp($1, "id") == 0)
                yyerror("To \"id\" δεν είναι τύπου table.");
            else if (strcmp($1, "divident") == 0)
                yyerror("To \"divident\" δεν είναι τύπου table.");
            else if (strcmp($1, "winners") == 0)
                yyerror("To \"winners\" δεν είναι τύπου table.");
            else if (strcmp($1, "distributed") == 0)
                yyerror("To \"distributed\" δεν είναι τύπου table.");
            else if (strcmp($1, "jackpot") == 0)
                yyerror("To \"jackpot\" δεν είναι τύπου table.");
            else if (strcmp($1, "fixed") == 0)
                yyerror("To \"fixed\" δεν είναι τύπου table.");
            else if (strcmp($1, "categoryType") == 0)
                yyerror("To \"categoryType\" δεν είναι τύπου table.");
            else if (strcmp($1, "gameType") == 0)
                yyerror("To \"gameType\" δεν είναι τύπου table.");
            else if (strcmp($1, "minimumDistributed") == 0)
                yyerror("To \"minimumDistributed\" δεν είναι τύπου table.");
            else if (strcmp($1, "columns") == 0)
                yyerror("To \"columns\" δεν είναι τύπου table.");
            else if (strcmp($1, "wagers") == 0)
                yyerror("To \"wagers\" δεν είναι τύπου table.");

            /* Έλεγχος άκυρων στοιχείων */
            else if (!((strcmp($1, "list") == 0) || (strcmp($1, "bonus") == 0)
|| (strcmp($1, "prizeCategories") == 0) || (strcmp($1, "addOn") == 0)))
                yyerror("Δεν υπάρχει αυτό το table στοιχείο.");
        }
        ;

table   : T_LBRACKET T_RBRACKET
        | T_LBRACKET integers T_RBRACKET
        | T_LBRACKET objects T_RBRACKET
        ;

objects : object    {$$ = $1;}
        | object T_COMMA objects    {$$ = $1;}
        ;

integers: T_INT {
```

```
            if ($1 < 1 || $1 > 45)
                yyerror("Κάποιο στοιχείο του \"list\" δεν ανήκει στο [1,45] ");
        }
        | T_INT T_COMMA integers {
            if ($1 < 1 || $1 > 45)
                yyerror("Κάποιο στοιχείο του \"list\" δεν ανήκει στο [1,45] ");

            if (list_counter == 4)
                yyerror("Το \"list\" περιέχει παραπάνω από 5 στοιχεία");
            list_counter++
        }
        ;

string  : T_QUOTE keyword_tokens T_QUOTE { $$ = $2;
            temp_val = $2;
            for (j = 0; j < i + 1; j++)
                if (strcmp(history[j], temp_val) != 0)
                    not_in_history = 1;
            if (not_in_history == 1) {
                strcpy(history[i++], temp_val);
                not_in_history = 0;
            }
            // printf("(i=%d)", i);
        }
        ;

keyword_tokens  : T_LAST {$$ = strdup(yytext);} | T_ACTIVE {$$ = strdup(yytext);}
                |  T_GAME_ID  {$$  =  strdup(yytext);}  |  T_DRAW_ID  {$$  =
strdup(yytext);}|  T_DRAW_TIME  {$$  =  strdup(yytext);}|  T_STATUS  {$$  =
strdup(yytext);}
                | T_DRAW_BREAK {$$ = strdup(yytext);} | T_VISUAL_DRAW {$$ =
strdup(yytext);} | T_PRICE_POINTS {$$ = strdup(yytext);}
                | T_WINNING_NUMBERS {$$ = strdup(yytext);} | T_PRIZE_CATEGORIES
{$$ = strdup(yytext);}
                | T_WAGER_STATISTICS {$$ = strdup(yytext);}
                |  T_AMOUNT  {$$  =  strdup(yytext);}|  T_LIST  {$$  =
strdup(yytext);}| T_BONUS {$$ = strdup(yytext);}| T_ID{$$ = strdup(yytext);}
                |  T_DIVIDENT  {$$  =  strdup(yytext);}|  T_WINNERS  {$$  =
strdup(yytext);}|  T_DISTRIBUTED  {$$  =  strdup(yytext);}|  T_JACKPOT  {$$  =
strdup(yytext);}| T_FIXED{$$ = strdup(yytext);}
                | T_CATEGORY_TYPE {$$ = strdup(yytext);}| T_GAME_TYPE {$$ =
strdup(yytext);}| T_MINIMUM_DISTRIBUTED{$$ = strdup(yytext);} | T_COLUMNS {$$ =
strdup(yytext);}|  T_WAGERS  {$$  =  strdup(yytext);}|  T_ADD_ON  {$$  =
strdup(yytext);} | T_WORD {$$ = strdup(yytext);}
                ;

%%
/* ========[ C ]======== */
```

```
int main(int argc, char **argv) {
    for (j = 0; j < HISTORY_SIZE; j++)
        history[j] = (char*)calloc(HISTORY_SIZE, sizeof(char));

    /* What's the input? */
    ++argv; --argc;
    if (argc > 0) {
        printf("Input from %s:\n", argv[0]);
        yyin = fopen(argv[0], "r");
    } else yyin = stdin;

    do {
        yyparse();
    } while (!feof(yyin));
    printf("\n");
    return 0;
}
void yyerror(char *s) {
    fprintf(stderr, "==[LINE %d: %s]==\n", yylineno/2+1, s);
}
```