**Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών**

**Τμήμα Μηχανικών Η/Υ & Πληροφορικής Πανεπιστήμιο Πατρών**

**Εργαστηριακή Άσκηση Εαρινού Εξαμήνου 2022**

Δούρου Βασιλική-Ευαγγελία, 1072633, 3o, [up1072633@upnet.gr](mailto:up1072633@upatras.gr)

Μπαλάφας Δημήτριος, 1072499, 3o, [up1072499@upnet.gr](mailto:up1072499@upatras.gr)

Πεσκελίδης Παύλος, 1072483, 3o, [up1072483@upnet.gr](mailto:up1072483@upatras.gr)

Ταμβάκης Θωμάς-Χρυσοβαλάντης, 1072631, 3o, [up1072631@upnet.gr](mailto:up1072631@upatras.gr)

Για τα τρία ερωτήματα της εργασίας αναπτύξαμε διαφορετικές γραμματικές της γλώσσας σε BNF. Τα τρία BNF παρατίθενται παρακάτω :

**Ερώτημα 1 :**

<start> ::= <FAA> <LASTW> <LAST> <C> <ACTIVEW> <ACTIVE> <FAD>

<last> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <WNW> <winningNumbers> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<active> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<winningNumbers> ::= <listW> <list> <bonusW> <bonus> <FAD> <C>

<listW> ::= <word> <AT> <AAA>

<bonusW> ::= <word> <AT> <AAA>

<prizeCatW> ::= <word> <AT> <AAA> <FAA>

<prizeCat> ::= <prizeCategories1> <FAD> <C> <Parray> <ADA> <C>

<Parray> ::= <FAA> <prizeCategories> <FAD> | <FAA> <prizeCategories> <FAD> <C> <Parray>

<prizeCategories> ::= <id> <divident> <winners> <distributed> <jackpot> <fixed> <categoryType> <gameType>

<prizeCategories1> ::= <prizeCategories> <C> <minimumDistributed>

<wagerStatistics> ::= <columns> <wagers> <addOn> <FAD>

<addOnAR> ::= <word> | <posI> | <negI> | <negR> | <posR>

<gameID> ::= <word> <AT> <posI> <C>

<drawID> ::= <word> <AT> <posI> <C>

<drawTime> ::= <word> <AT> <posI> <C>

<drawBreak> ::= <word> <AT> <posI> <C>

<visualDraw> ::= <word> <AT> <posI> <C>

<columns> ::= <word> <AT> <posI> <C>

<wagers> ::= <word> <AT> <posI> <C>

<winners> ::= <word> <AT> <posI> <C>

<posI> ::= (0-9)+

<categoryType> ::= <word> <AT> <posI> <C>

<status> ::= <word> <AT> <word> <C>

<gameType> ::= <word> <AT> <word>

<word> ::= '"' (\\.|(^"\\))\* '"'

<id> ::= <word> <AT> <posI> <C>

<pricePoints> ::= <amount> <FAD> <C>

<list> ::= <larray> <ADA> <C>

<larray> ::= <posI> | <posI> <C> <larray>

<bonus> ::= <posI> <ADA>

<amount> ::= <word> <AT> <posR>

<divident> ::= <word> <AT> <posR> <C>

<distributed> ::= <word> <AT> <posR> <C>

<jackpot> ::= <word> <AT> <posR> <C>

<fixed> ::= <word> <AT> <posR> <C>

<minimumDistributed> ::= <word> <AT> <posR>

<posR> ::= (0-9)+.(0-9)+

<addOn> ::= <word> <AT> <AAA> <addOnEXP> <ADA> | <word> <AT> <AAA> <ADA>

<negI> ::= -(0-9)+

<negR> ::= -(0-9)+.(0-9)+

<PPW> ::= <word> <AT> <FAA>

<WNW> ::= <word> <AT> <FAA>

<WSW> ::= <word> <AT> <FAA>

<activeW> ::= <word> <AT> <FAA>

<lastw> ::= <WORD> <AT> <FAA>

<addOnEXP> ::= <addOnAR> | <addOnAR> <C> <addOnEXP>

<addOnAR> ::= <word> | <posI> | <negI> | <negR> | <posR>

<C> ::= ","

<FAA> ::= "{"

<FAD> ::= "}"

<AT> ::= ":"

<AAA> ::= "["

<ADA> ::= "]"

/////Επιπρόσθετοι κανόνες που δεν συνδέονται άμεσα με το BNF :

///// \n {}

///// [ \t]+ /\*ignore whitespace\*/

///// . { yyerror("Undefined symbol"); exit(1); }

**Ερώτημα 2 :**

<start> ::= <FAA> <LASTW> <LAST> <C> <ACTIVEW> <ACTIVE> <FAD> | <startR>

<startR> ::= <FAA> <contentW> <content> <totalPages> <totalElements> <tLast> <numberOfElements> <sortW> <sort> <tFirst> <size> <number> <FAD>

<contentW> ::= <word> <AT> <AAA>

<contentAR> ::= <FAA> <last> | <FAA> <last> <C> <contentAR>

<content> ::= <contentAR> <ADA> <C>

<last> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <WNW> <winningNumbers> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<active> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<winningNumbers> ::= <listW> <list> <bonusW> <bonus> <FAD> <C>

<prizeCat> ::= <prizeCategories1> <FAD> <C> <Parray> <ADA> <C>

<Parray> ::= <FAA> <prizeCategories> <FAD> | <FAA> <prizeCategories> <FAD> <C> <Parray>

<prizeCategories> ::= <id> <divident> <winners> <distributed> <jackpot> <fixed> <categoryType> <gameType>

<prizeCategories1> ::= <prizeCategories> <C> <minimumDistributed>

<wagerStatistics> ::= <columns> <wagers> <addOn> <FAD>

<addOnAR> ::= <word> | <posI> | <negI> | <negR> | <posR>

<gameID> ::= <word> <AT> <posI> <C>

<drawID> ::= <word> <AT> <posI> <C>

<drawTime> ::= <word> <AT> <posI> <C>

<drawBreak> ::= <word> <AT> <posI> <C>

<visualDraw> ::= <word> <AT> <posI> <C>

<columns> ::= <word> <AT> <posI> <C>

<wagers> ::= <word> <AT> <posI> <C>

<winners> ::= <word> <AT> <posI> <C>

<posI> ::= (0-9)+

<categoryType> ::= <word> <AT> <posI> <C>

<status> ::= <word> <AT> <word> <C>

<gameType> ::= <word> <AT> <word>

<word> ::= '"' (\\.|(^"\\))\* '"'

<id> ::= <word> <AT> <posI> <C>

<pricePoints> ::= <amount> <FAD> <C>

<list> ::= <larray> <ADA> <C>

<larray> ::= <posI> | <posI> <C> <larray>

<bonus> ::= <posI> <ADA>

<amount> ::= <word> <AT> <posR>

<divident> ::= <word> <AT> <posR> <C>

<distributed> ::= <word> <AT> <posR> <C>

<jackpot> ::= <word> <AT> <posR> <C>

<fixed> ::= <word> <AT> <posR> <C>

<minimumDistributed> ::= <word> <AT> <posR>

<posR> ::= (0-9)+.(0-9)+

<addOn> ::= <word> <AT> <AAA> <addOnEXP> <ADA> | <word> <AT> <AAA> <ADA>

<negI> ::= -(0-9)+

<negR> ::= -(0-9)+.(0-9)+

<PPW> ::= <word> <AT> <FAA>

<WNW> ::= <word> <AT> <FAA>

<WSW> ::= <word> <AT> <FAA>

<activeW> ::= <word> <AT> <FAA>

<lastw> ::= <WORD> <AT> <FAA>

<listW> ::= <word> <AT> <AAA>

<bonusW> ::= <word> <AT> <AAA>

<prizeCatW> ::= <word> <AT> <AAA> <FAA>

<addOnEXP> ::= <addOnAR> | <addOnAR> <C> <addOnEXP>

<totalPages> ::= <word> <AT> <posI> <C>

<totalElements> ::= <word> <AT> <posI> <C>

<tLast> ::= <word> <AT> <bool> <C>

<numberOfElements> ::= <word> <AT> <posI> <C>

<direction> ::= <word> <AT> <word> <C>

<property> ::= <word> <AT> <word> <C>

<ignoreCase> ::= <word> <AT> <bool> <C>

<nullHandling> ::= <word> <AT> <word> <C>

<descending> ::= <word> <AT> <bool> <C>

<ascending> ::= <word> <AT> <bool>

<sortW> ::= <word> <AT> <AAA>

<sort> ::= <FAA> <direction> <property> <ignoreCase> <nullHandling> <descending> <ascending> <FAD> <ADA> <C>

<tFirst> ::= <word> <AT> <bool> <C>

<size> ::= <word> <AT> <posI> <C>

<number> ::= <word> <AT> <posI>

<bool> ::= "true" | "false"

<C> ::= ","

<FAA> ::= "{"

<FAD> ::= "}"

<AT> ::= ":"

<AAA> ::= "["

<ADA> ::= "]"

/////Επιπρόσθετοι κανόνες που δεν συνδέονται άμεσα με το BNF :

///// \n {}

///// [ \t]+ /\*ignore whitespace\*/

///// . { yyerror("Undefined symbol"); exit(1); }

**Ερώτημα 3 :**

<start> ::= <FAA> <LASTW> <LAST> <C> <ACTIVEW> <ACTIVE> <FAD> | <startR>

<startR> ::= <FAA> <contentW> <content> <totalPages> <totalElements> <tLast> <numberOfElements> <sortW> <sort> <tFirst> <size> <number> <FAD>

<contentW> ::= <word> <AT> <AAA>

<contentAR> ::= <FAA> <last> | <FAA> <last> <C> <contentAR>

<content> ::= <contentAR> <ADA> <C>

<last> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <WNW> <winningNumbers> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<active> ::= <gameID> <drawID> <drawTime> <status> <drawBreak> <visualDraw> <PPW> <pricePoints> <prizeCatW> <prizeCat> <WSW> <wagerStatistics> <FAD>

<winningNumbers> ::= <listW> <list> <bonusW> <bonus> <FAD> <C>

<prizeCat> ::= <prizeCategories1> <FAD> <C> <Parray> <ADA> <C>

<Parray> ::= <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD> <C> <FAA> <prizeCategories> <FAD>

<prizeCategories> ::= <id> <divident> <winners> <distributed> <jackpot> <fixed> <categoryType> <gameType>

<prizeCategories1> ::= <prizeCategories> <C> <minimumDistributed>

<wagerStatistics> ::= <columns> <wagers> <addOn> <FAD>

<addOnAR> ::= <word> | <posI> | <negI> | <negR> | <posR>

<gameID> ::= <word> <AT> <posI> <C>

<drawID> ::= <word> <AT> <posI> <C>

<drawTime> ::= <word> <AT> <posI> <C>

<drawBreak> ::= <word> <AT> <posI> <C>

<visualDraw> ::= <word> <AT> <posI> <C>

<columns> ::= <word> <AT> <posI> <C>

<wagers> ::= <word> <AT> <posI> <C>

<winners> ::= <word> <AT> <posI> <C>

<posI> ::= (0-9)+

<categoryType> ::= <word> <AT> <posI> <C>

<status> ::= <word> <AT> <word> <C>

<gameType> ::= <word> <AT> <word>

<word> ::= '"' (\\.|(^"\\))\* '"'

<id> ::= <word> <AT> <posI> <C>

<pricePoints> ::= <amount> <FAD> <C>

<list> ::= <larray> <ADA> <C>

<larray> ::= <posI> <C> <posI> <C> <posI> <C> <posI> <C> <posI>

<bonus> ::= <posI> <ADA>

<amount> ::= <word> <AT> <posR>

<divident> ::= <word> <AT> <posR> <C>

<distributed> ::= <word> <AT> <posR> <C>

<jackpot> ::= <word> <AT> <posR> <C>

<fixed> ::= <word> <AT> <posR> <C>

<minimumDistributed> ::= <word> <AT> <posR>

<posR> ::= (0-9)+.(0-9)+

<addOn> ::= <word> <AT> <AAA> <addOnEXP> <ADA> | <word> <AT> <AAA> <ADA>

<negI> ::= -(0-9)+

<negR> ::= -(0-9)+.(0-9)+

<PPW> ::= <word> <AT> <FAA>

<WNW> ::= <word> <AT> <FAA>

<WSW> ::= <word> <AT> <FAA>

<activeW> ::= <word> <AT> <FAA>

<lastw> ::= <WORD> <AT> <FAA>

<listW> ::= <word> <AT> <AAA>

<bonusW> ::= <word> <AT> <AAA>

<prizeCatW> ::= <word> <AT> <AAA> <FAA>

<addOnEXP> ::= <addOnAR> | <addOnAR> <C> <addOnEXP>

<totalPages> ::= <word> <AT> <posI> <C>

<totalElements> ::= <word> <AT> <posI> <C>

<tLast> ::= <word> <AT> <bool> <C>

<numberOfElements> ::= <word> <AT> <posI> <C>

<direction> ::= <word> <AT> <word> <C>

<property> ::= <word> <AT> <word> <C>

<ignoreCase> ::= <word> <AT> <bool> <C>

<nullHandling> ::= <word> <AT> <word> <C>

<descending> ::= <word> <AT> <bool> <C>

<ascending> ::= <word> <AT> <bool>

<sortW> ::= <word> <AT> <AAA>

<sort> ::= <FAA> <direction> <property> <ignoreCase> <nullHandling> <descending> <ascending> <FAD> <ADA> <C>

<tFirst> ::= <word> <AT> <bool> <C>

<size> ::= <word> <AT> <posI> <C>

<number> ::= <word> <AT> <posI>

<bool> ::= "true" | "false"

<C> ::= ","

<FAA> ::= "{"

<FAD> ::= "}"

<AT> ::= ":"

<AAA> ::= "["

<ADA> ::= "]"

/////Επιπρόσθετοι κανόνες που δεν συνδέονται άμεσα με το BNF :

///// \n {}

///// [ \t]+ /\*ignore whitespace\*/

///// . { yyerror("Undefined symbol"); exit(1); }

Tα τελικά αρχεία περιγραφής της γλώσσας που δίνονται ως είσοδος στα Flex και Bison είναι τα εξής :

**Ερώτημα 1**

* **FLEX :**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

#include "dok.tab.h"

extern int yyerror(char \*s);

%}

%option noyywrap

%option yylineno

%%

"," {yylval.ch=strdup(yytext);return C;}

"{" {yylval.ch=strdup(yytext);return FAA;}

"}" {yylval.ch=strdup(yytext);return FAD;}

":" {yylval.ch=strdup(yytext);return AT;}

"[" {yylval.ch=strdup(yytext);return AAA;}

"]" {yylval.ch=strdup(yytext);return ADA;}

[0-9]+ {yylval.IN=atoi(yytext);return POSI;}

[-][0-9]+ {yylval.IN=atoi(yytext);return NEGI;}

[0-9]+[.][0-9]+ {yylval.F=atof(yytext);return POSR;}

[-][0-9]+[.][0-9]+ {yylval.F=atof(yytext);return NEGR;}

\"(\\.|[^"\\])\*\" {yylval.ch=strdup(yytext);return WORD;}

\n {}

[ \t]+ /\*ignore whitespace\*/

. { yyerror("Undefined symbol"); exit(1); }

%%

* **BISON**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

extern int yyval;

extern FILE \*yyin;

extern FILE \*yyout;

int yylex();

extern int yylineno;

int errf=0;

%}

%locations

%union {

char \*ch;

int IN;

float F;

}

%token <ch> FAA

%token <ch> FAD

%token <ch> AT

%token <ch> AAA

%token <ch> ADA

%token <ch> C

%token <IN> POSI

%token <F> POSR

%token <IN> NEGI

%token <F> NEGR

%token <ch> WORD

%%

START: FAA LASTW LAST C ACTIVEW ACTIVE FAD;

LASTW: WORD AT FAA {char target[]={'"','l','a','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LAST: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS WNW WINNINGNUMBERS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

ACTIVEW: WORD AT FAA {char target[]={'"','a','c','t','i','v','e','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ACTIVE: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

GAMEID: WORD AT POSI C {char target[]={'"','g','a','m','e','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWID: WORD AT POSI C{char target[]={'"','d','r','a','w','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWTIME: WORD AT POSI C {char target[]={'"','d','r','a','w','T','i','m','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

STATUS: WORD AT WORD C {char target[]={'"','s','t','a','t','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWBREAK: WORD AT POSI C {char target[]={'"','d','r','a','w','B','r','e','a','k','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

VISUALDRAW: WORD AT POSI C {char target[]={'"','v','i','s','u','a','l','D','r','a','w','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PPW: WORD AT FAA{char target[]={'"','p','r','i','c','e','P','o','i','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRICEPOINTS: AMOUNT FAD C ;

AMOUNT: WORD AT POSR {char target[]={'"','a','m','o','u','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WNW: WORD AT FAA {char target[]={'"','w','i','n','n','i','n','g','N','u','m','b','e','r','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WINNINGNUMBERS: LISTW LIST BONUSW BONUS FAD C ;

LISTW: WORD AT AAA {char target[]={'"','l','i','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LIST: LARRAY ADA C ;

LARRAY: POSI| POSI C LARRAY;

BONUSW: WORD AT AAA {char target[]={'"','b','o','n','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

BONUS: POSI ADA ;

PRIZECATW: WORD AT AAA FAA{char target[]={'"','p','r','i','z','e','C','a','t','e','g','o','r','i','e','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRIZECAT: PRIZECATEGORIES1 FAD C PARRAY ADA C;

PARRAY: FAA PRIZECATEGORIES FAD| FAA PRIZECATEGORIES FAD C PARRAY;

PRIZECATEGORIES1: PRIZECATEGORIES C MINIMUMDISTRIBUTED;

PRIZECATEGORIES: ID DIVIDENT WINNERS DISTRIBUTED JACKPOT FIXED CATEGORYTYPE GAMETYPE;

ID: WORD AT POSI C {char target[]={'"','i','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};if(($3!=1)&&($3!=2)&&($3!=3)&&($3!=4)&&($3!=5)&&($3!=6)&&($3!=7)&&($3!=8)){yyerror(":wrong sequence");YYERROR;};};

DIVIDENT: WORD AT POSR C{char target[]={'"','d','i','v','i','d','e','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WINNERS: WORD AT POSI C{char target[]={'"','w','i','n','n','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DISTRIBUTED: WORD AT POSR C {char target[]={'"','d','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

JACKPOT: WORD AT POSR C {char target[]={'"','j','a','c','k','p','o','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

FIXED: WORD AT POSR C {char target[]={'"','f','i','x','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

CATEGORYTYPE: WORD AT POSI C {char target[]={'"','c','a','t','e','g','o','r','y','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

GAMETYPE: WORD AT WORD {char target[]={'"','g','a','m','e','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

MINIMUMDISTRIBUTED: WORD AT POSR {char target[]={'"','m','i','n','i','m','u','m','D','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WSW: WORD AT FAA {char target[]={'"','w','a','g','e','r','S','t','a','t','i','s','t','i','c','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WAGERSTATISTICS: COLUMNS WAGERS ADDON FAD ;

COLUMNS: WORD AT POSI C {char target[]={'"','c','o','l','u','m','n','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WAGERS: WORD AT POSI C {char target[]={'"','w','a','g','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

ADDON: WORD AT AAA ADDONEXP ADA | WORD AT AAA ADA {char target[]={'"','a','d','d','O','n','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ADDONEXP: ADDONAR| ADDONAR C ADDONEXP;

ADDONAR: WORD| POSI| NEGI| NEGR| POSR;

%%

void yyerror(char \*s)

{

errf=1;

printf("Error in line %d. Type of error %s\n", yylineno,s);

}

int main ()

{

char ch;

yyin=fopen("last\_result.json","r");

ch=fgetc(yyin);

while(ch!=EOF){

printf("%c",ch);

ch=fgetc(yyin);

}

fclose(yyin);

yyin = fopen("last\_result.json","r");

yyparse ();

fclose(yyin);

if(errf==0)

{

printf("The input is correct.\n");

}

return 0;

}

**Ερώτημα 2**

* **FLEX :**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

#include "dok2.tab.h"

extern int yyerror(char \*s);

%}

%option noyywrap

%option yylineno

%%

"," {yylval.ch=strdup(yytext);return C;}

"{" {yylval.ch=strdup(yytext);return FAA;}

"}" {yylval.ch=strdup(yytext);return FAD;}

":" {yylval.ch=strdup(yytext);return AT;}

"[" {yylval.ch=strdup(yytext);return AAA;}

"]" {yylval.ch=strdup(yytext);return ADA;}

[0-9]+ {yylval.IN=atoi(yytext);return POSI;}

[-][0-9]+ {yylval.IN=atoi(yytext);return NEGI;}

[0-9]+[.][0-9]+ {yylval.F=atof(yytext);return POSR;}

[-][0-9]+[.][0-9]+ {yylval.F=atof(yytext);return NEGR;}

"true"|"false" {yylval.ch=strdup(yytext); return BOOL;}

\"(\\.|[^"\\])\*\" {yylval.ch=strdup(yytext);return WORD;}

\n {}

[ \t]+ {}

. { yyerror("Undefined symbol"); exit(1); }

%%

* **BISON**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

extern int yyval;

extern FILE \*yyin;

extern FILE \*yyout;

int yylex();

extern int yylineno;

int errf=0;

%}

%locations

%union {

char \*ch;

int IN;

float F;

}

%token <ch> FAA

%token <ch> FAD

%token <ch> AT

%token <ch> AAA

%token <ch> ADA

%token <ch> C

%token <IN> POSI

%token <F> POSR

%token <IN> NEGI

%token <F> NEGR

%token <ch> BOOL

%token <ch> WORD

%%

START: FAA LASTW LAST C ACTIVEW ACTIVE FAD|STARTR;

STARTR:FAA CONTENTW CONTENT TOTALPAGES TOTALELEMENTS TLAST NUMBEROFELEMENTS SORTW SORT TFIRST SIZE NUMBER FAD;

CONTENTW: WORD AT AAA {char target[]={'"','c','o','n','t','e','n','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

CONTENTAR: FAA LAST| FAA LAST C CONTENTAR;

CONTENT: CONTENTAR ADA C;

LASTW: WORD AT FAA {char target[]={'"','l','a','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LAST: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS WNW WINNINGNUMBERS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

ACTIVEW: WORD AT FAA {char target[]={'"','a','c','t','i','v','e','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ACTIVE: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

GAMEID: WORD AT POSI C {char target[]={'"','g','a','m','e','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWID: WORD AT POSI C{char target[]={'"','d','r','a','w','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWTIME: WORD AT POSI C {char target[]={'"','d','r','a','w','T','i','m','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

STATUS: WORD AT WORD C {char target[]={'"','s','t','a','t','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWBREAK: WORD AT POSI C {char target[]={'"','d','r','a','w','B','r','e','a','k','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

VISUALDRAW: WORD AT POSI C {char target[]={'"','v','i','s','u','a','l','D','r','a','w','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PPW: WORD AT FAA{char target[]={'"','p','r','i','c','e','P','o','i','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRICEPOINTS: AMOUNT FAD C ;

AMOUNT: WORD AT POSR {char target[]={'"','a','m','o','u','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WNW: WORD AT FAA {char target[]={'"','w','i','n','n','i','n','g','N','u','m','b','e','r','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WINNINGNUMBERS: LISTW LIST BONUSW BONUS FAD C ;

LISTW: WORD AT AAA {char target[]={'"','l','i','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LIST: LARRAY ADA C ;

LARRAY: POSI| POSI C LARRAY;

BONUSW: WORD AT AAA {char target[]={'"','b','o','n','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

BONUS: POSI ADA ;

PRIZECATW: WORD AT AAA FAA{char target[]={'"','p','r','i','z','e','C','a','t','e','g','o','r','i','e','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRIZECAT: PRIZECATEGORIES1 FAD C PARRAY ADA C;

PARRAY: FAA PRIZECATEGORIES FAD| FAA PRIZECATEGORIES FAD C PARRAY;

PRIZECATEGORIES1: PRIZECATEGORIES C MINIMUMDISTRIBUTED;

PRIZECATEGORIES: ID DIVIDENT WINNERS DISTRIBUTED JACKPOT FIXED CATEGORYTYPE GAMETYPE;

ID: WORD AT POSI C {char target[]={'"','i','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};if(($3!=1)&&($3!=2)&&($3!=3)&&($3!=4)&&($3!=5)&&($3!=6)&&($3!=7)&&($3!=8)){yyerror(":wrong sequence");YYERROR;};};

DIVIDENT: WORD AT POSR C{char target[]={'"','d','i','v','i','d','e','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WINNERS: WORD AT POSI C{char target[]={'"','w','i','n','n','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DISTRIBUTED: WORD AT POSR C {char target[]={'"','d','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

JACKPOT: WORD AT POSR C {char target[]={'"','j','a','c','k','p','o','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

FIXED: WORD AT POSR C {char target[]={'"','f','i','x','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

CATEGORYTYPE: WORD AT POSI C {char target[]={'"','c','a','t','e','g','o','r','y','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

GAMETYPE: WORD AT WORD {char target[]={'"','g','a','m','e','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

MINIMUMDISTRIBUTED: WORD AT POSR {char target[]={'"','m','i','n','i','m','u','m','D','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WSW: WORD AT FAA {char target[]={'"','w','a','g','e','r','S','t','a','t','i','s','t','i','c','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WAGERSTATISTICS: COLUMNS WAGERS ADDON FAD ;

COLUMNS: WORD AT POSI C {char target[]={'"','c','o','l','u','m','n','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WAGERS: WORD AT POSI C {char target[]={'"','w','a','g','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

ADDON: WORD AT AAA ADDONEXP ADA | WORD AT AAA ADA {char target[]={'"','a','d','d','O','n','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ADDONEXP: ADDONAR| ADDONAR C ADDONEXP;

ADDONAR: WORD| POSI| NEGI| NEGR| POSR;

TOTALPAGES: WORD AT POSI C {char target[]={'"','t','o','t','a','l','P','a','g','e','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

TOTALELEMENTS: WORD AT POSI C {char target[]={'"','t','o','t','a','l','E','l','e','m','e','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

TLAST: WORD AT BOOL C {char target[]={'"','l','a','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

NUMBEROFELEMENTS: WORD AT POSI C {char target[]={'"','n','u','m','b','e','r','O','f','E','l','e','m','e','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DIRECTION: WORD AT WORD C {char target[]={'"','d','i','r','e','c','t','i','o','n','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PROPERTY: WORD AT WORD C {char target[]={'"','p','r','o','p','e','r','t','y','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

IGNORECASE: WORD AT BOOL C {char target[]={'"','i','g','n','o','r','e','C','a','s','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

NULLHANDLING: WORD AT WORD C {char target[]={'"','n','u','l','l','H','a','n','d','l','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DESCENDING: WORD AT BOOL C {char target[]={'"','d','e','s','c','e','n','d','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

ASCENDING: WORD AT BOOL {char target[]={'"','a','s','c','e','n','d','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

SORTW: WORD AT AAA {char target[]={'"','s','o','r','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

SORT: FAA DIRECTION PROPERTY IGNORECASE NULLHANDLING DESCENDING ASCENDING FAD ADA C;

TFIRST: WORD AT BOOL C {char target[]={'"','f','i','r','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

SIZE: WORD AT POSI C{char target[]={'"','s','i','z','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

NUMBER: WORD AT POSI {char target[]={'"','n','u','m','b','e','r','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

%%

void yyerror(char \*s)

{

errf=1;

printf("Error in line %d. Type of error %s\n", yylineno,s);

}

int main ()

{

char ch;

yyin=fopen("testrange.json","r");

ch=fgetc(yyin);

while(ch!=EOF){

printf("%c",ch);

ch=fgetc(yyin);

}

fclose(yyin);

yyin = fopen("testrange.json","r");

yyparse ();

fclose(yyin);

if(errf==0)

{

printf("The input is correct.\n");

}

return 0;

}

**Ερώτημα 3**

* **FLEX :**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

#include "y.tab.h"

extern int yyerror(char \*s);

%}

%option noyywrap

%option yylineno

%%

"," {yylval.ch=strdup(yytext);return C;}

"{" {yylval.ch=strdup(yytext);return FAA;}

"}" {yylval.ch=strdup(yytext);return FAD;}

":" {yylval.ch=strdup(yytext);return AT;}

"[" {yylval.ch=strdup(yytext);return AAA;}

"]" {yylval.ch=strdup(yytext);return ADA;}

[0-9]+ {yylval.IN=atoi(yytext);return POSI;}

[-][0-9]+ {yylval.IN=atoi(yytext);return NEGI;}

[0-9]+[.][0-9]+ {yylval.F=atof(yytext);return POSR;}

[-][0-9]+[.][0-9]+ {yylval.F=atof(yytext);return NEGR;}

"true"|"false" {yylval.ch=strdup(yytext); return BOOL;}

\"(\\.|[^"\\])\*\" {yylval.ch=strdup(yytext);return WORD;}

\n {}

[ \t]+ {}

. { yyerror("Undefined symbol"); exit(1); }

%%

* **BISON**

%{

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

extern int yyval;

extern FILE \*yyin;

extern FILE \*yyout;

int yylex();

extern int yylineno;

int errf=0;

%}

%locations

%union {

char \*ch;

int IN;

float F;

}

%token <ch> FAA

%token <ch> FAD

%token <ch> AT

%token <ch> AAA

%token <ch> ADA

%token <ch> C

%token <IN> POSI

%token <F> POSR

%token <IN> NEGI

%token <F> NEGR

%token <ch> BOOL

%token <ch> WORD

%%

START: FAA LASTW LAST C ACTIVEW ACTIVE FAD|STARTR;

STARTR:FAA CONTENTW CONTENT TOTALPAGES TOTALELEMENTS TLAST NUMBEROFELEMENTS SORTW SORT TFIRST SIZE NUMBER FAD;

CONTENTW: WORD AT AAA {char target[]={'"','c','o','n','t','e','n','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

CONTENTAR: FAA LAST| FAA LAST C CONTENTAR;

CONTENT: CONTENTAR ADA C;

LASTW: WORD AT FAA {char target[]={'"','l','a','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LAST: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS WNW WINNINGNUMBERS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

ACTIVEW: WORD AT FAA {char target[]={'"','a','c','t','i','v','e','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ACTIVE: GAMEID DRAWID DRAWTIME STATUS DRAWBREAK VISUALDRAW PPW PRICEPOINTS PRIZECATW PRIZECAT WSW WAGERSTATISTICS FAD ;

GAMEID: WORD AT POSI C {char target[]={'"','g','a','m','e','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;}; if(($3!=1100)&&($3!=1110)&&($3!=2100)&&($3!=2101)&&($3!=5103)&&($3!=5104)&&($3!=5106)){yyerror(":wrong selection");YYERROR;};};

DRAWID: WORD AT POSI C{char target[]={'"','d','r','a','w','I','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWTIME: WORD AT POSI C {char target[]={'"','d','r','a','w','T','i','m','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

STATUS: WORD AT WORD C {char target[]={'"','s','t','a','t','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DRAWBREAK: WORD AT POSI C {char target[]={'"','d','r','a','w','B','r','e','a','k','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

VISUALDRAW: WORD AT POSI C {char target[]={'"','v','i','s','u','a','l','D','r','a','w','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PPW: WORD AT FAA{char target[]={'"','p','r','i','c','e','P','o','i','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRICEPOINTS: AMOUNT FAD C ;

AMOUNT: WORD AT POSR {char target[]={'"','a','m','o','u','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WNW: WORD AT FAA {char target[]={'"','w','i','n','n','i','n','g','N','u','m','b','e','r','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WINNINGNUMBERS: LISTW LIST BONUSW BONUS FAD C ;

LISTW: WORD AT AAA {char target[]={'"','l','i','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

LIST: LARRAY ADA C ;

LARRAY: POSI C POSI C POSI C POSI C POSI {if(($1<1 || $1>45)||($3<1 || $3>45)||($5<1 || $5>45)||($7<1 || $7>45)||($9<1 || $9>45)){yyerror(":wrong value");YYERROR;};};

BONUSW: WORD AT AAA {char target[]={'"','b','o','n','u','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

BONUS: POSI ADA ;

PRIZECATW: WORD AT AAA FAA{char target[]={'"','p','r','i','z','e','C','a','t','e','g','o','r','i','e','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PRIZECAT: PRIZECATEGORIES1 FAD C PARRAY ADA C;

PARRAY: FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD C FAA PRIZECATEGORIES FAD;

PRIZECATEGORIES1: PRIZECATEGORIES C MINIMUMDISTRIBUTED;

PRIZECATEGORIES: ID DIVIDENT WINNERS DISTRIBUTED JACKPOT FIXED CATEGORYTYPE GAMETYPE;

ID: WORD AT POSI C {char target[]={'"','i','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};if(($3!=1)&&($3!=2)&&($3!=3)&&($3!=4)&&($3!=5)&&($3!=6)&&($3!=7)&&($3!=8)){yyerror(":wrong sequence");YYERROR;};};

DIVIDENT: WORD AT POSR C{char target[]={'"','d','i','v','i','d','e','n','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WINNERS: WORD AT POSI C{char target[]={'"','w','i','n','n','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DISTRIBUTED: WORD AT POSR C {char target[]={'"','d','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

JACKPOT: WORD AT POSR C {char target[]={'"','j','a','c','k','p','o','t','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

FIXED: WORD AT POSR C {char target[]={'"','f','i','x','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

CATEGORYTYPE: WORD AT POSI C {char target[]={'"','c','a','t','e','g','o','r','y','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

GAMETYPE: WORD AT WORD {char target[]={'"','g','a','m','e','T','y','p','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

MINIMUMDISTRIBUTED: WORD AT POSR {char target[]={'"','m','i','n','i','m','u','m','D','i','s','t','r','i','b','u','t','e','d','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WSW: WORD AT FAA {char target[]={'"','w','a','g','e','r','S','t','a','t','i','s','t','i','c','s','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

WAGERSTATISTICS: COLUMNS WAGERS ADDON FAD ;

COLUMNS: WORD AT POSI C {char target[]={'"','c','o','l','u','m','n','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

WAGERS: WORD AT POSI C {char target[]={'"','w','a','g','e','r','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

ADDON: WORD AT AAA ADDONEXP ADA | WORD AT AAA ADA {char target[]={'"','a','d','d','O','n','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

ADDONEXP: ADDONAR| ADDONAR C ADDONEXP;

ADDONAR: WORD| POSI| NEGI| NEGR| POSR;

TOTALPAGES: WORD AT POSI C {char target[]={'"','t','o','t','a','l','P','a','g','e','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

TOTALELEMENTS: WORD AT POSI C {char target[]={'"','t','o','t','a','l','E','l','e','m','e','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

TLAST: WORD AT BOOL C {char target[]={'"','l','a','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

NUMBEROFELEMENTS: WORD AT POSI C {char target[]={'"','n','u','m','b','e','r','O','f','E','l','e','m','e','n','t','s','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DIRECTION: WORD AT WORD C {char target[]={'"','d','i','r','e','c','t','i','o','n','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

PROPERTY: WORD AT WORD C {char target[]={'"','p','r','o','p','e','r','t','y','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

IGNORECASE: WORD AT BOOL C {char target[]={'"','i','g','n','o','r','e','C','a','s','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

NULLHANDLING: WORD AT WORD C {char target[]={'"','n','u','l','l','H','a','n','d','l','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

DESCENDING: WORD AT BOOL C {char target[]={'"','d','e','s','c','e','n','d','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

ASCENDING: WORD AT BOOL {char target[]={'"','a','s','c','e','n','d','i','n','g','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

SORTW: WORD AT AAA {char target[]={'"','s','o','r','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

SORT: FAA DIRECTION PROPERTY IGNORECASE NULLHANDLING DESCENDING ASCENDING FAD ADA C;

TFIRST: WORD AT BOOL C {char target[]={'"','f','i','r','s','t','"','\0'};if((strcmp($1,target)!=0)){yyerror(":wrong sequence");YYERROR;};};

SIZE: WORD AT POSI C{char target[]={'"','s','i','z','e','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

NUMBER: WORD AT POSI {char target[]={'"','n','u','m','b','e','r','"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};};

%%

void yyerror(char \*s)

{

errf=1;

printf("Error in line %d. Type of error %s\n", yylineno,s);

}

int main ()

{

char ch;

yyin=fopen("testrange.json","r");

ch=fgetc(yyin);

while(ch!=EOF){

printf("%c",ch);

ch=fgetc(yyin);

}

fclose(yyin);

yyin = fopen("testrange.json","r");

yyparse ();

fclose(yyin);

if(errf==0)

{

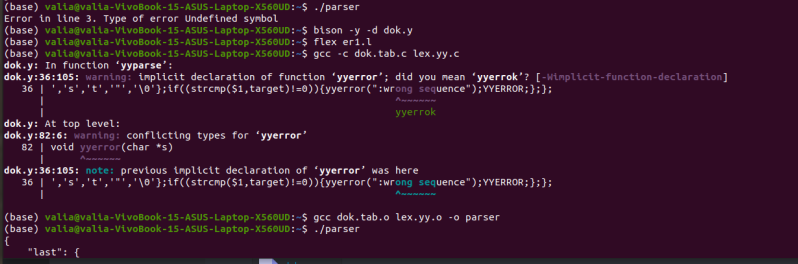
printf("The input is correct.\n");

}

return 0;

}

Οι εντολές εκτέλεσης των αρχείων FLEX και BISON εμφανίζουν τα εξής :

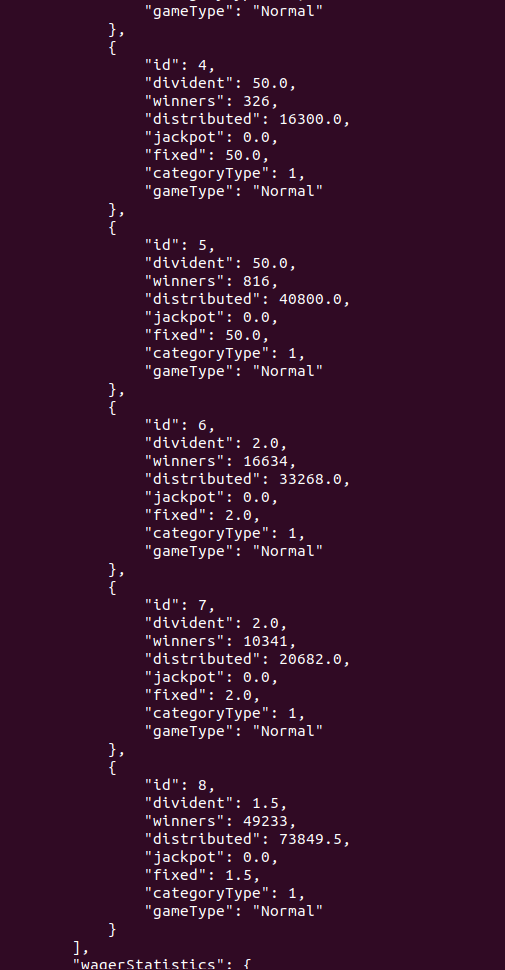


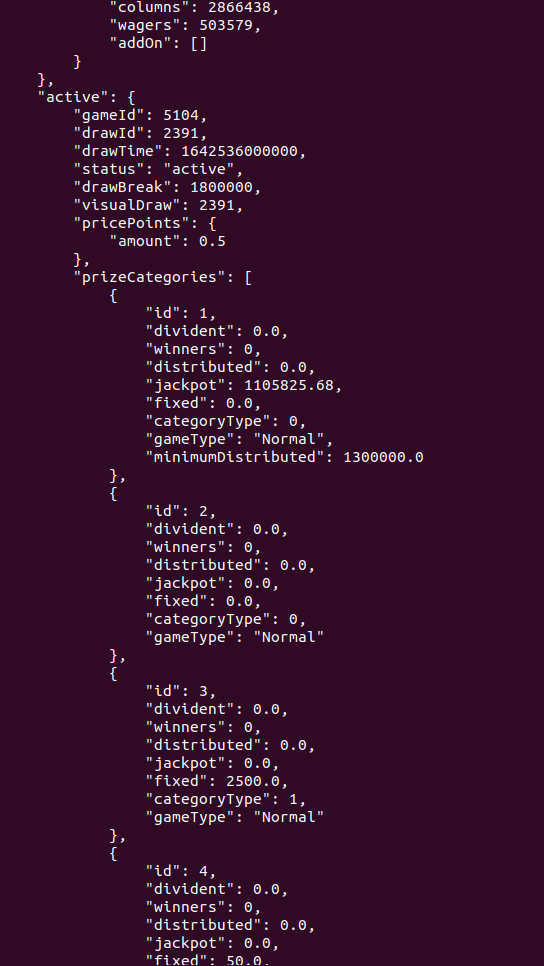
Παρακάτω παρατίθενται screenshots παραδειγμάτων εφαρμογής :

**Ερώτημα 1 :**

**H επιτυχημένη εκτέλεση του ερωτήματος 1 εμφανίζει τα παρακάτω :**



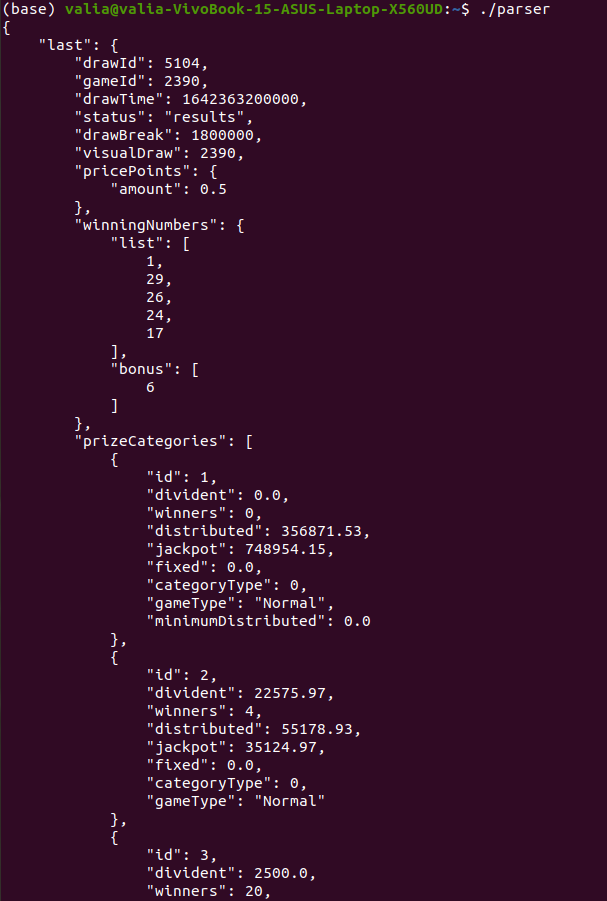


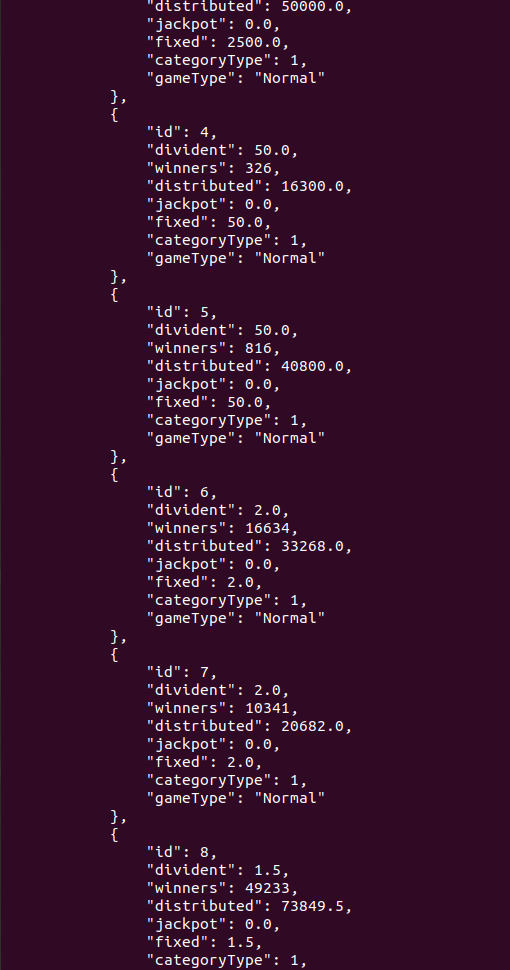


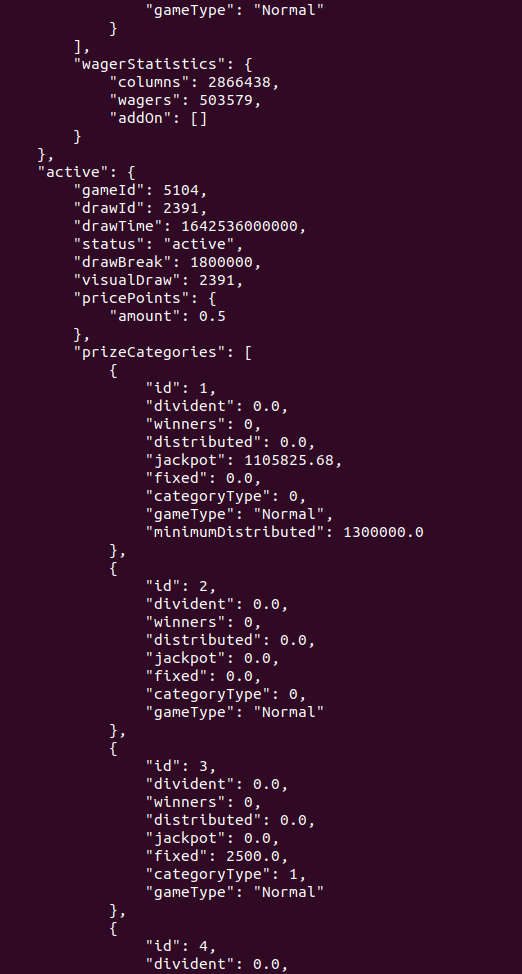


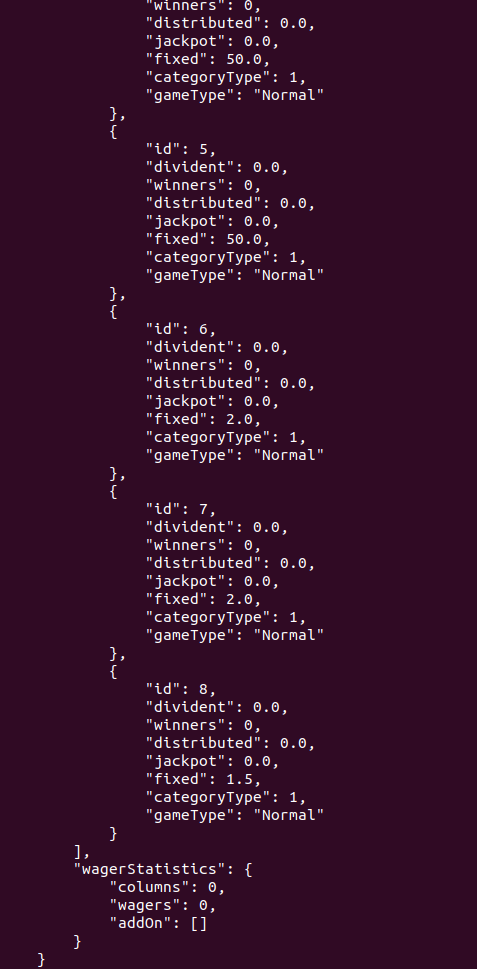
Στο τέλος της εξόδου εμφανίζεται κατάλληλο μήνυμα που επιβεβαιώνει ότι η είσοδος είναι σωστή.

**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Λανθασμένη Σειρά**





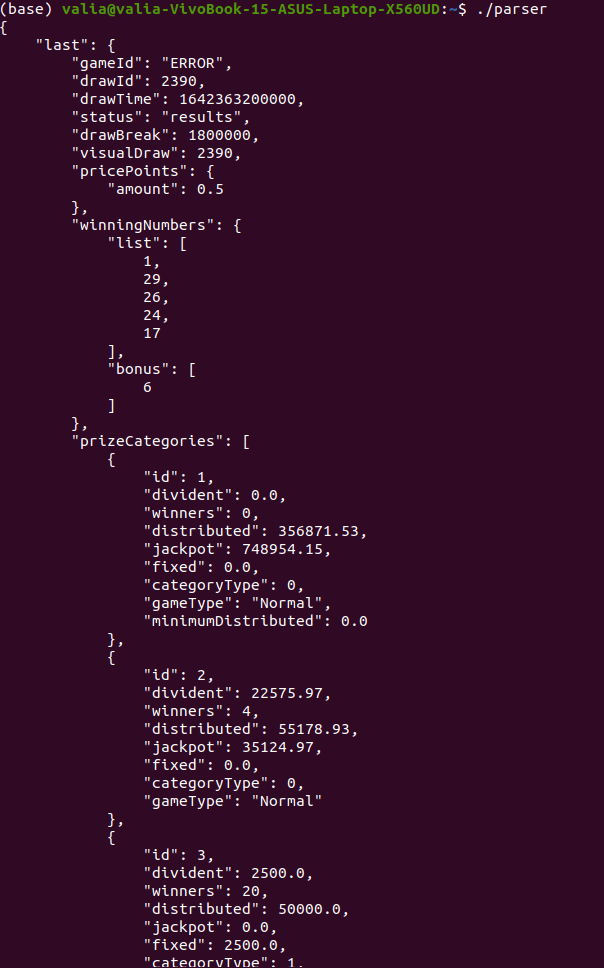


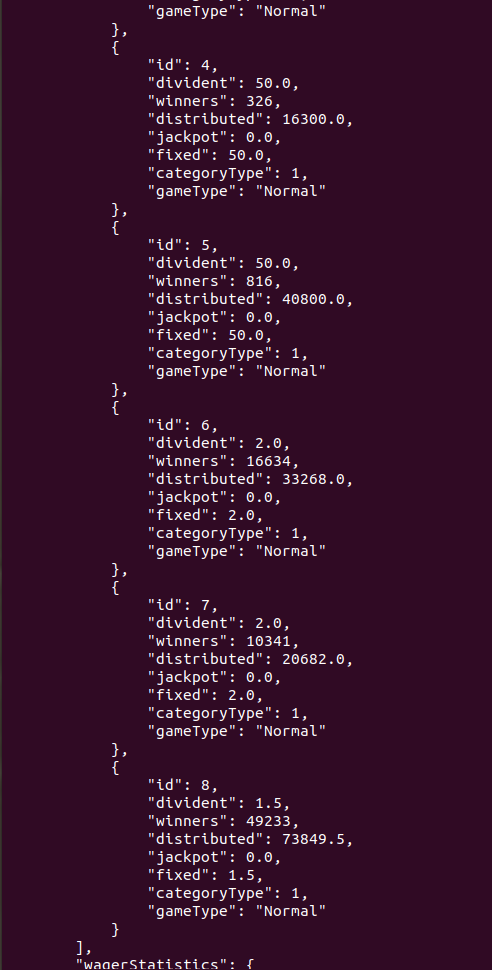


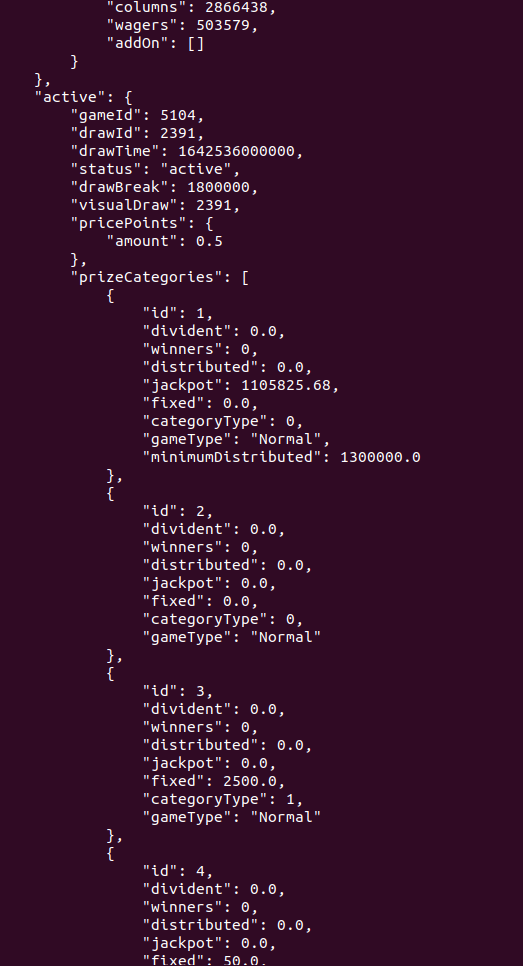


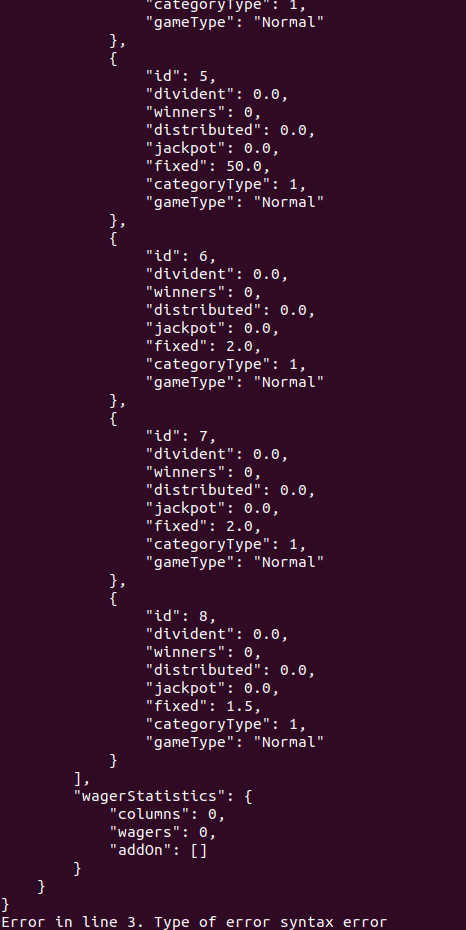
Παρατηρούμε πως αν αλλάξουμε τη σειρά των πρώτων δύο πεδίων, δηλαδή αν βάλουμε στη θέση του gameId το drawId, εμφανίζεται μήνυμα πως υπάρχει σφάλμα στη γραμμή 3. Αυτό συμβαίνει επειδή, ακόμη κι αν τεχνικά υπάρχει σφάλμα και στη γραμμή 4, το πρώτο σφάλμα εντοπίστηκε στην γραμμή 3 (στο πεδίο gameId) και εκεί σταματάει η εκτέλεση.

**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Syntax error**







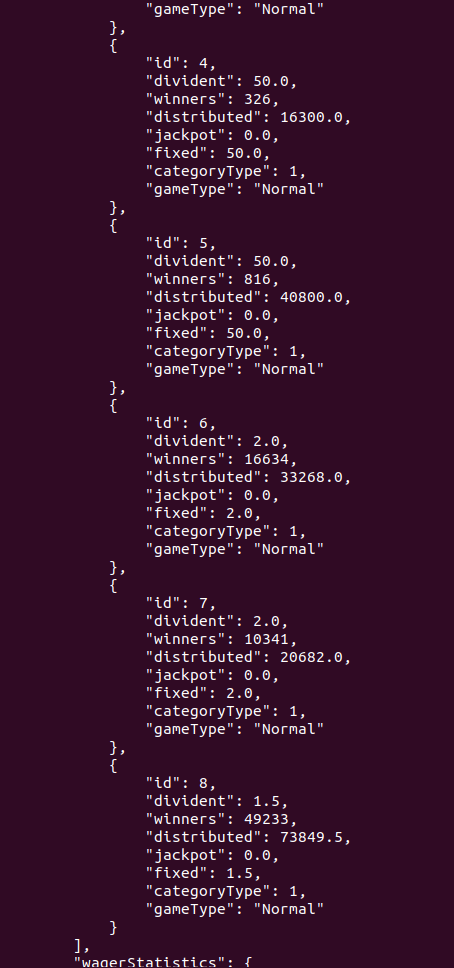


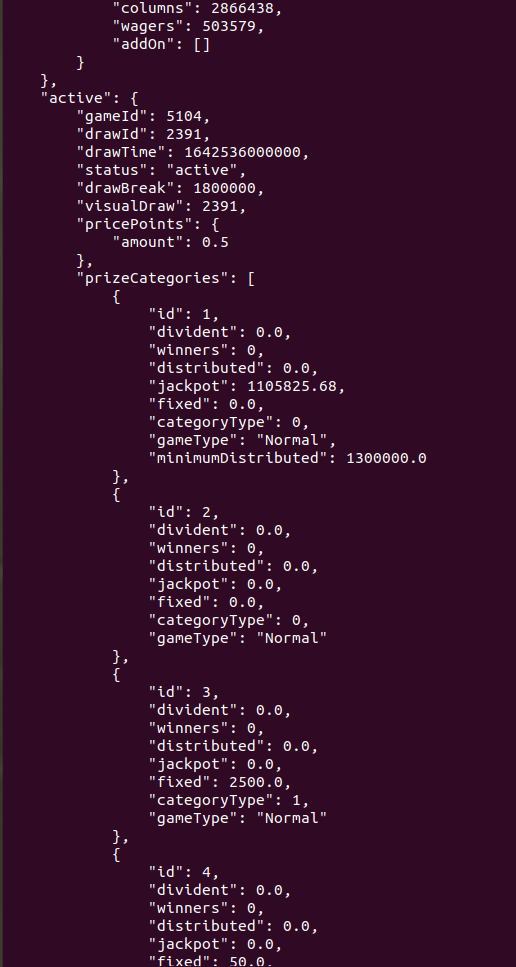
Σε αυτήν την περίπτωση για να προκαλέσουμε σφάλμα, αντικαταστήσαμε τη τιμή του πεδίου gameId με τη λέξη ERROR. Επομένως, ενώ περιμέναμε στην είσοδο τιμή τύπου int λάβαμε string.

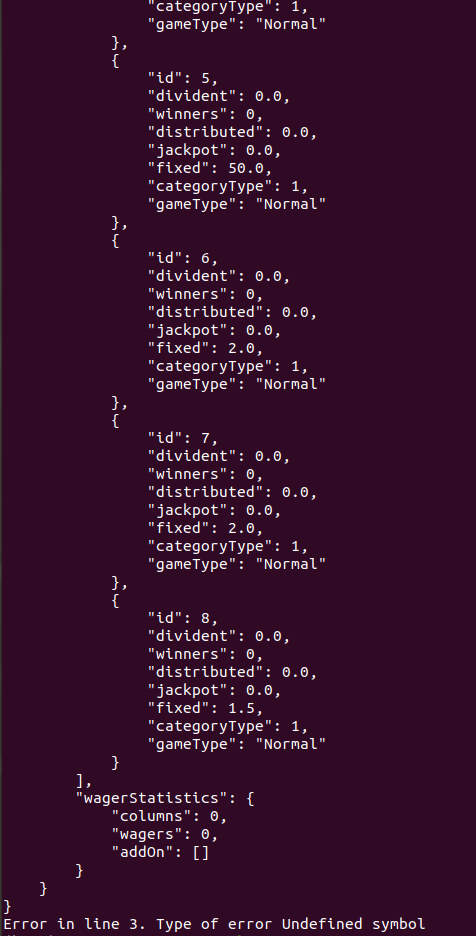
Έτσι, δημιουργείται σφάλμα τύπου syntax και εκτυπώνεται το αντίστοιχο μήνυμα στο τέλος της εισόδου.

**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : undefined symbol**







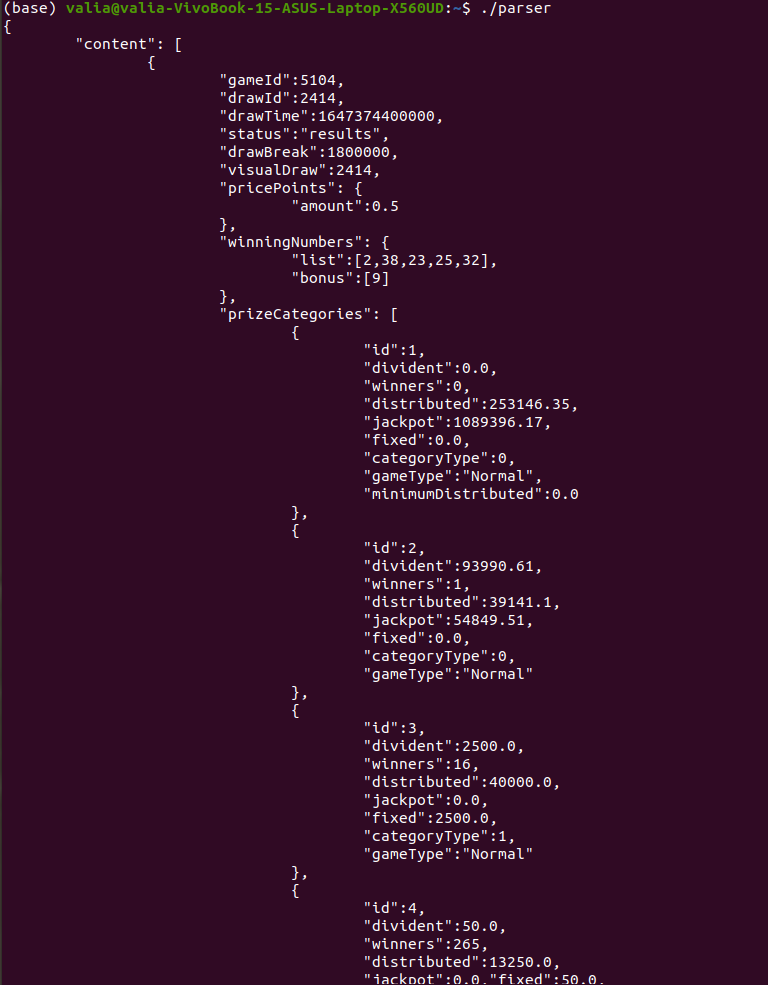


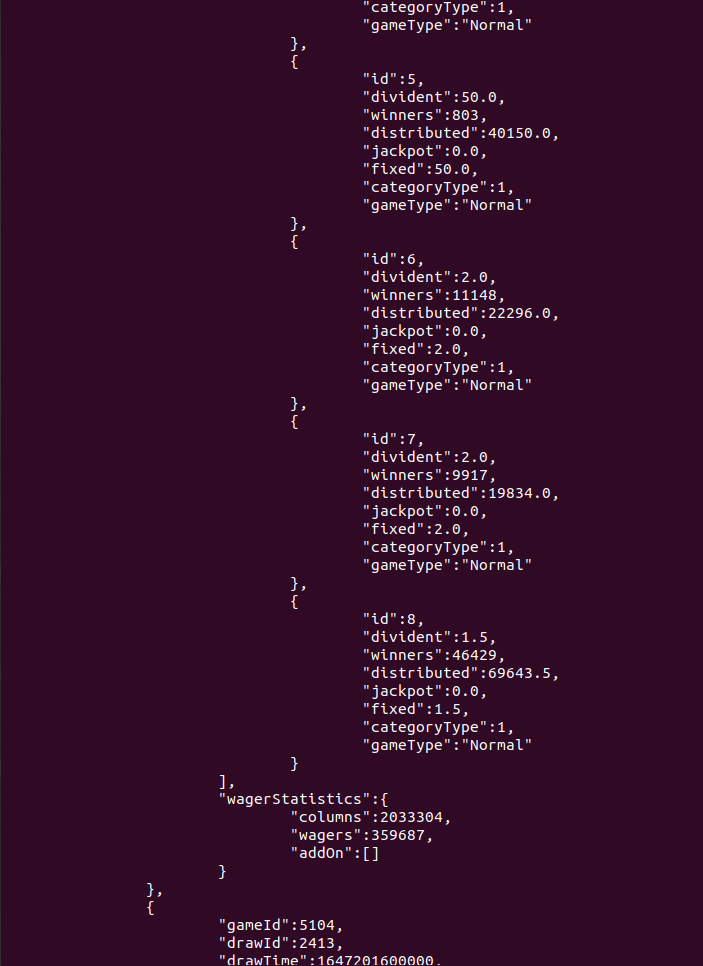
Για το σφάλμα undefined symbol, προσθέσαμε στη τιμή του πεδίου gameId ένα σύμβολο που δεν είναι ακέραιος αριθμός. Έτσι, λαμβάνοντας στην είσοδο ένα ελληνικό “α”, εκτός από τον αναμενόμενο ακέραιο, προκαλεί την εκτύπωση του αντίστοιχου μηνύματος σφάλματος.

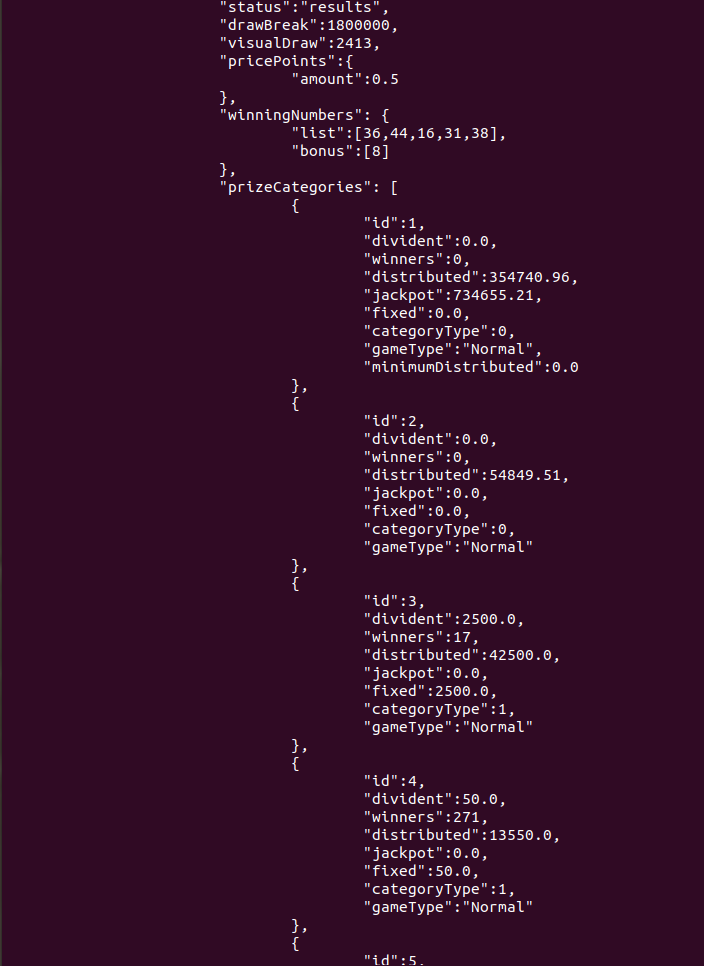
**Ερώτημα 2 :**

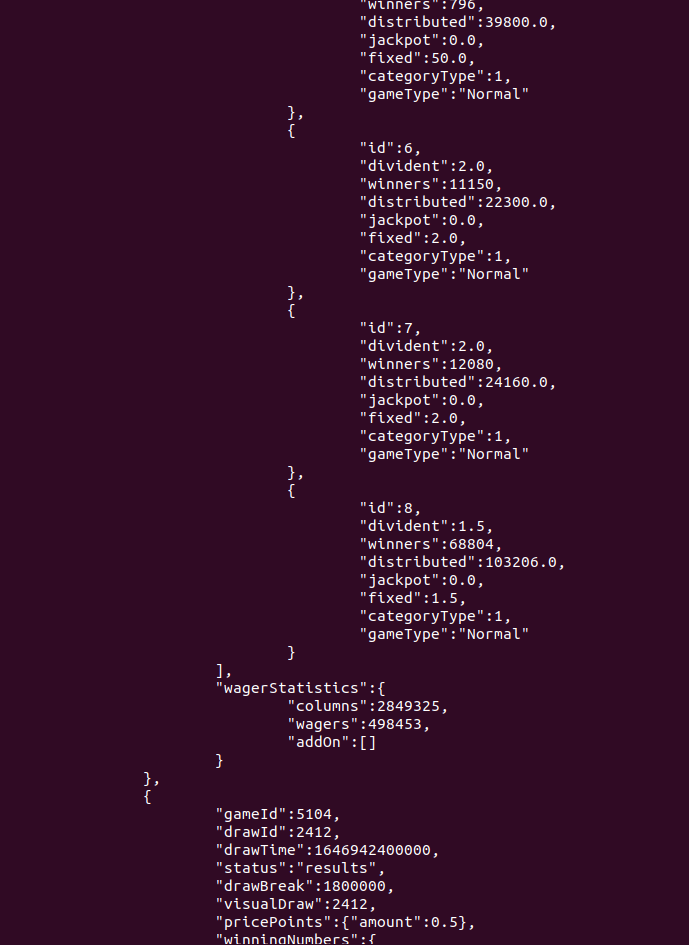
**H επιτυχημένη εκτέλεση του ερωτήματος 2 για το αρχείο last\_result εμφανίζει την ίδια έξοδο με την εκτέλεση του ερωτήματος 1.**

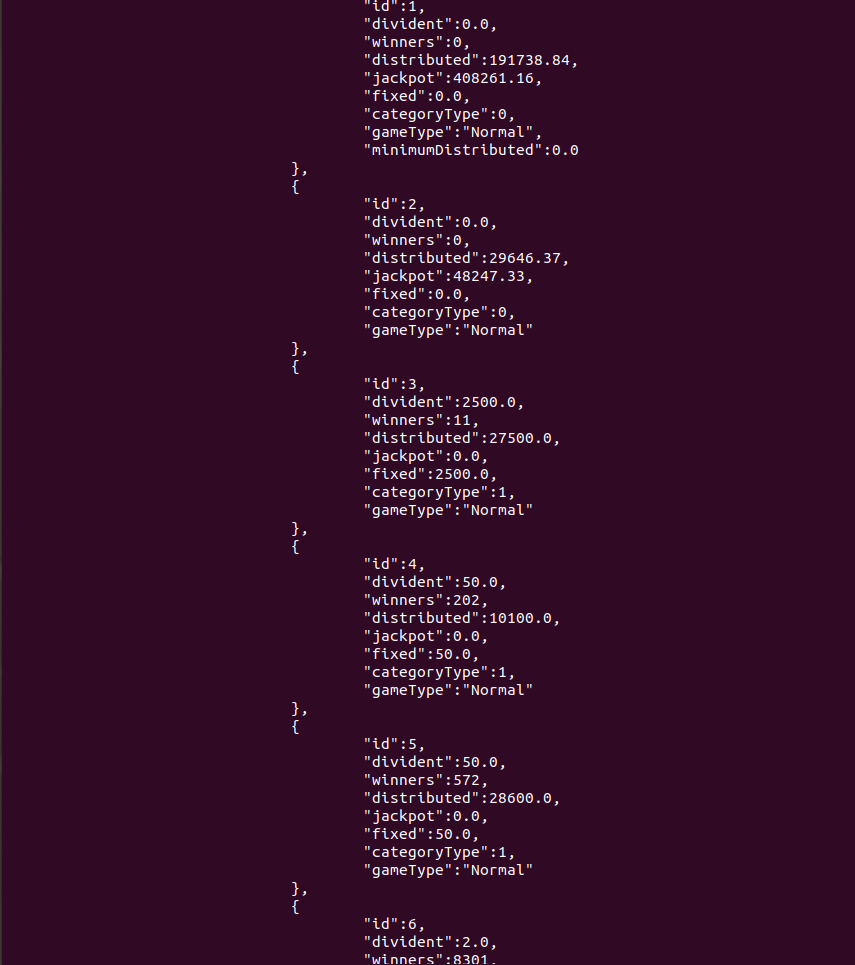
**H επιτυχημένη εκτέλεση του ερωτήματος 2 για το αρχείο range\_result εμφανίζει τα παρακάτω :**

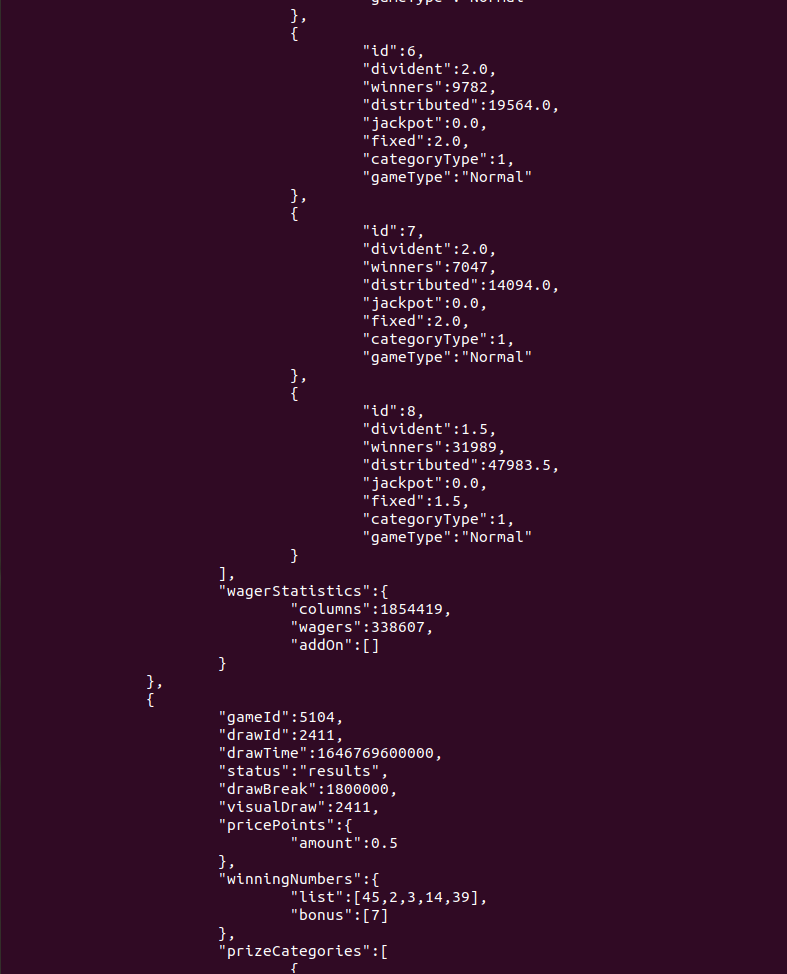


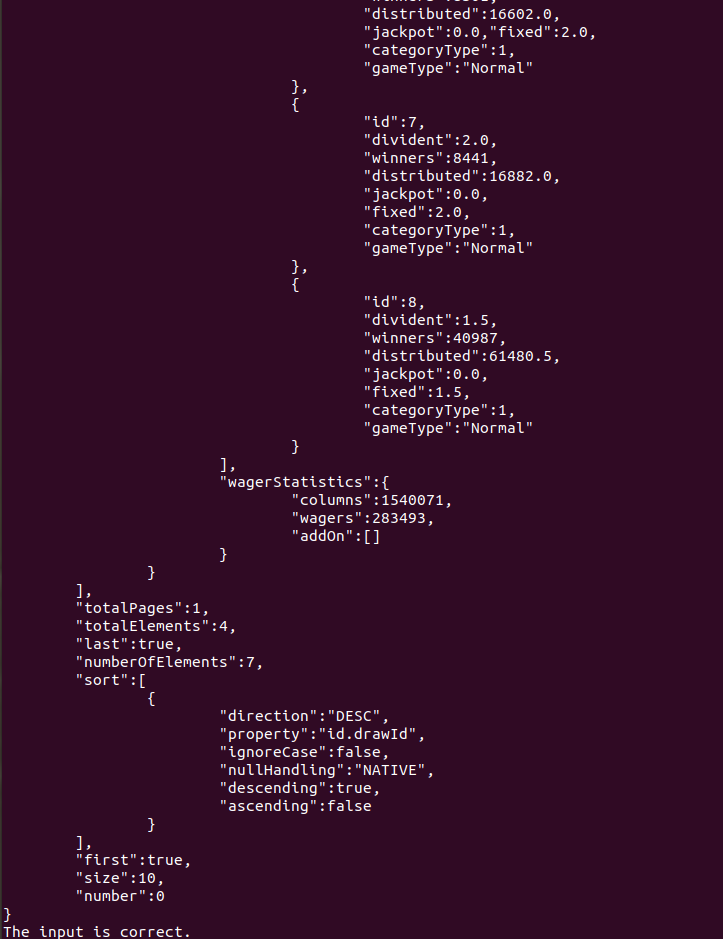






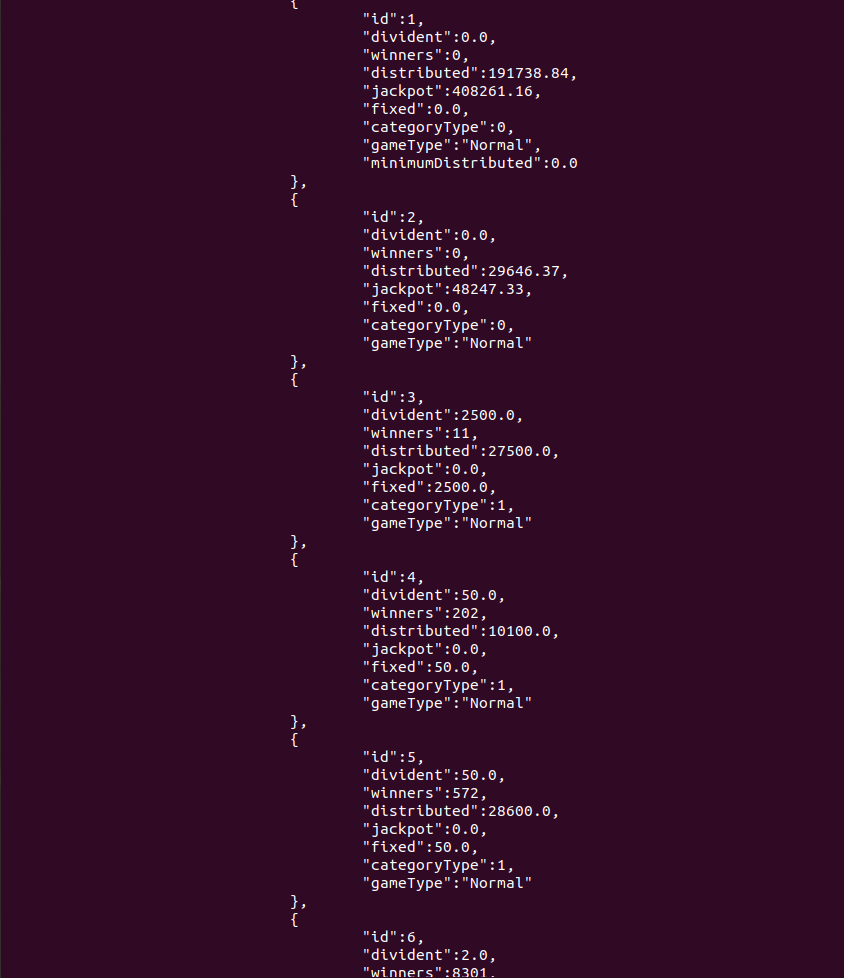
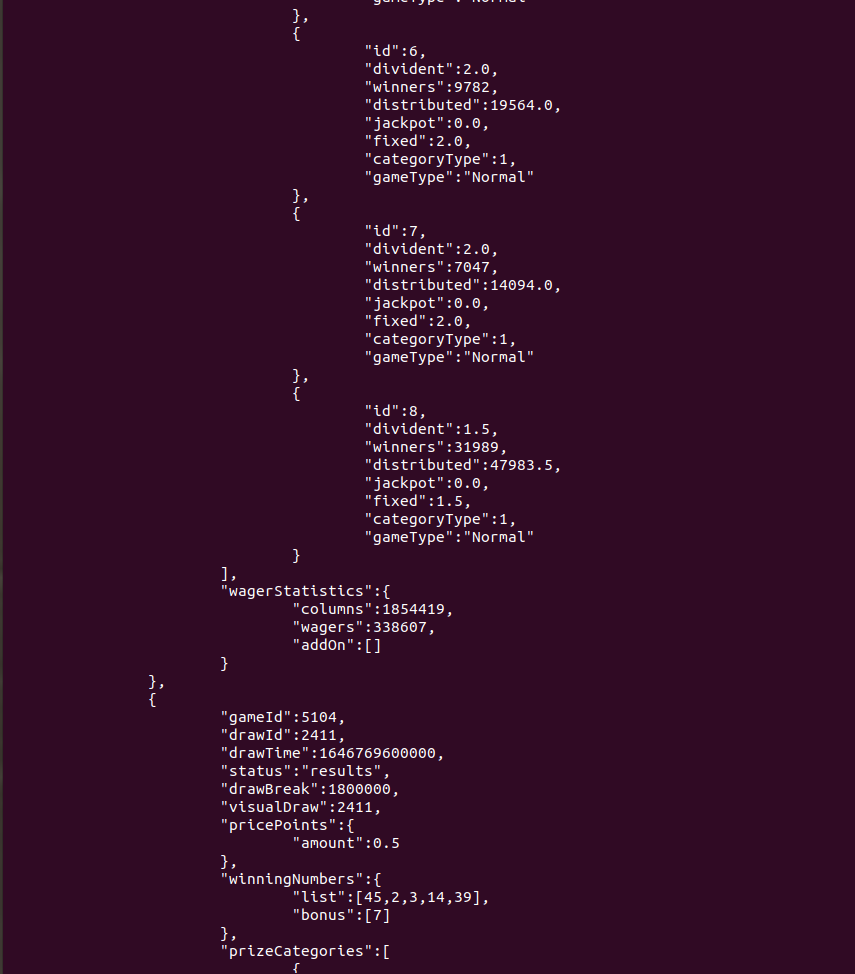
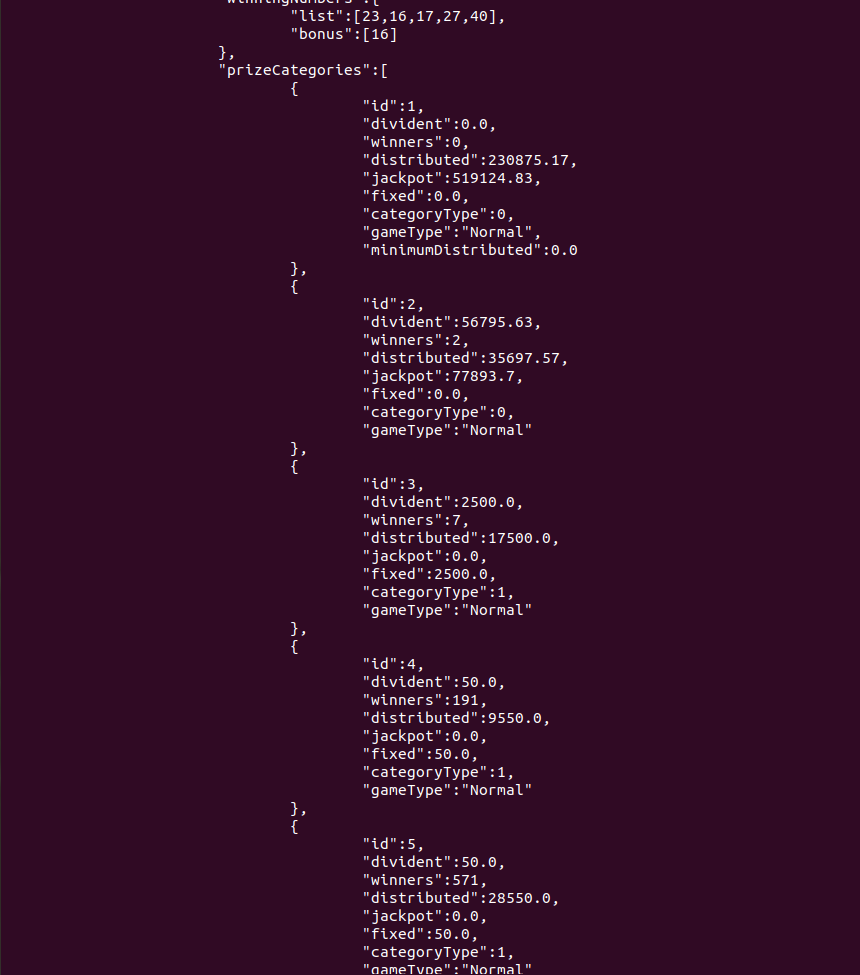
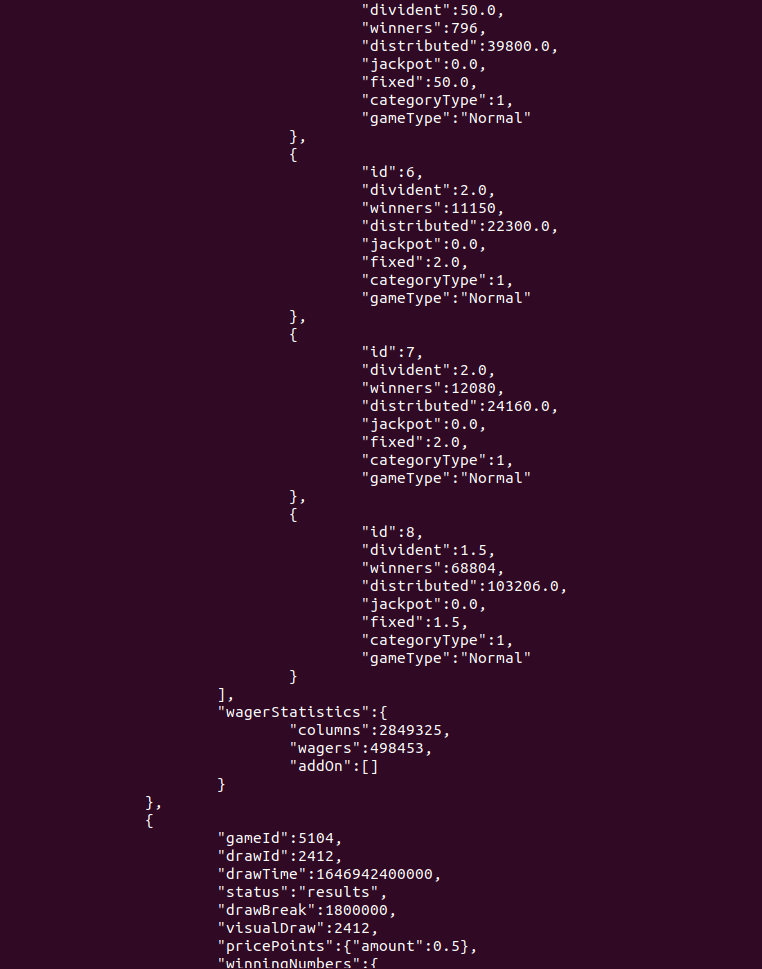
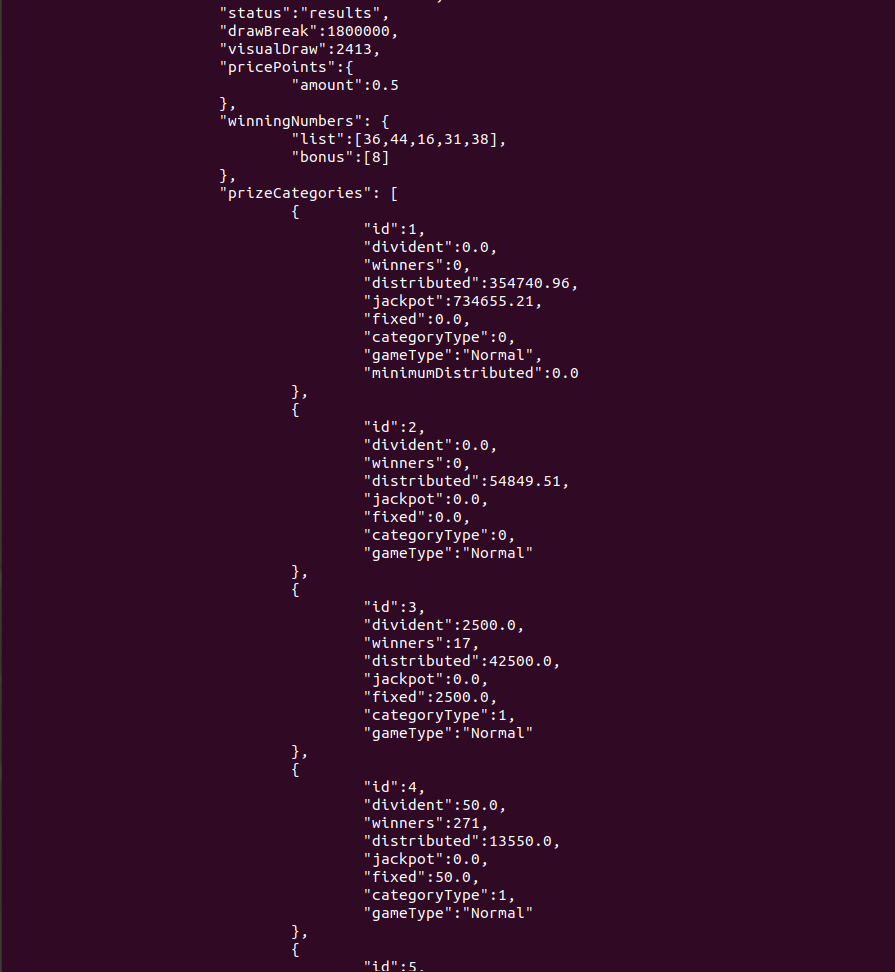
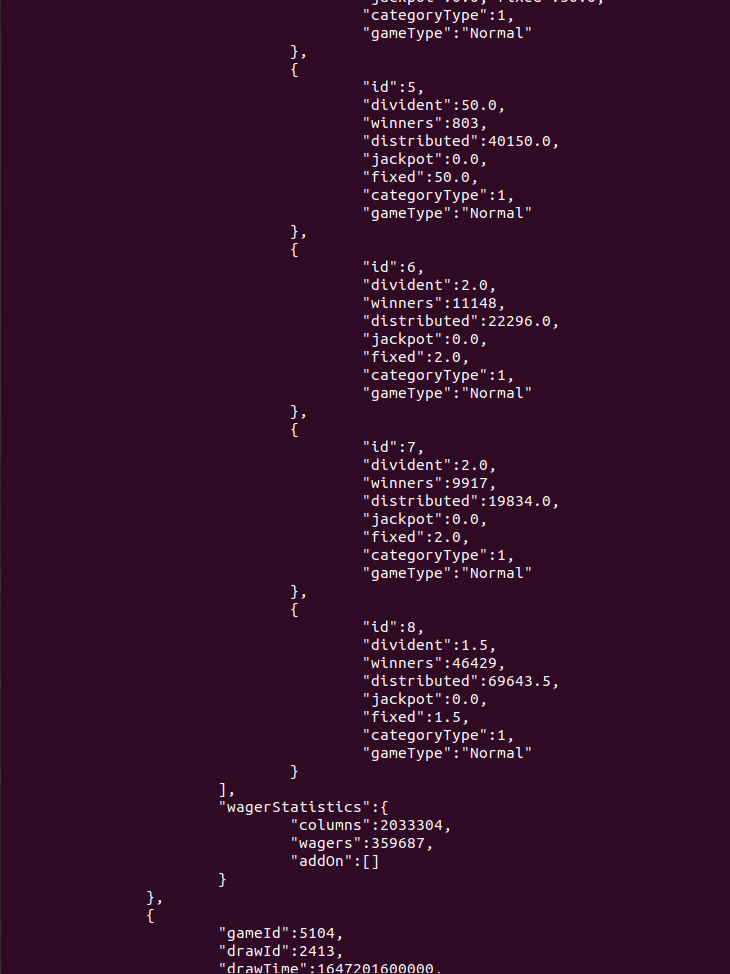
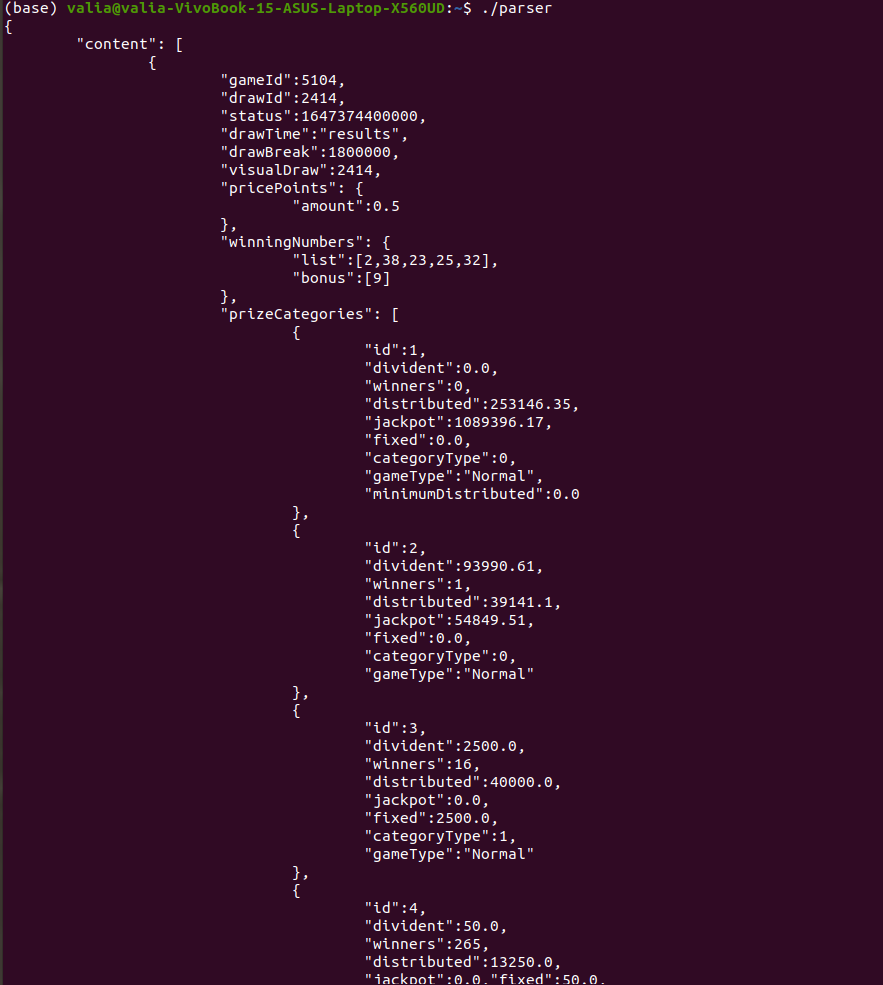


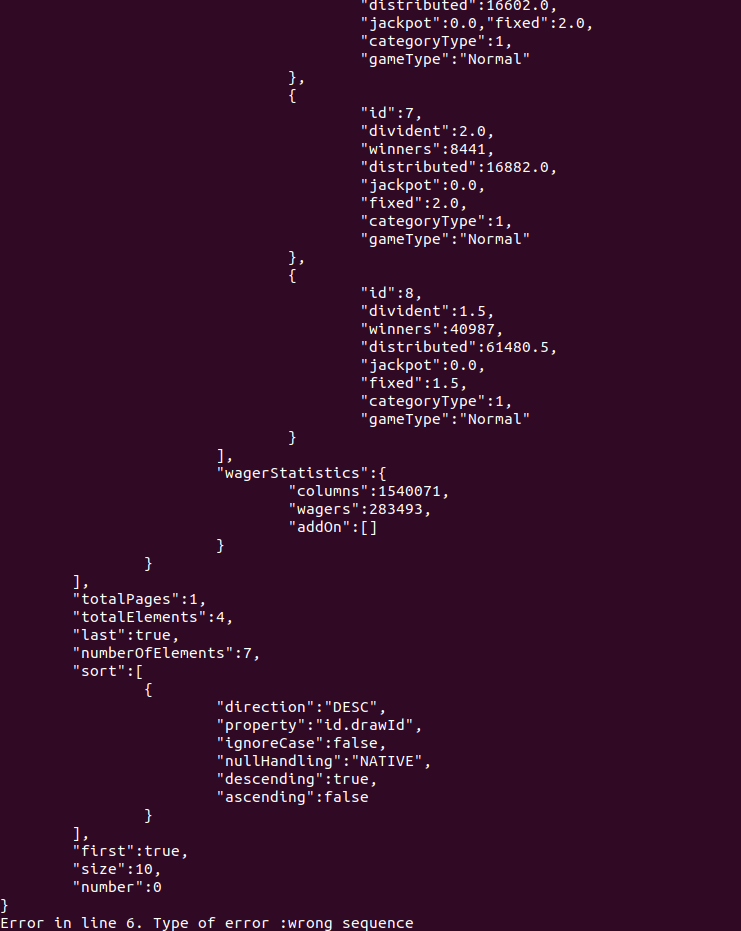




Στο τέλος της εξόδου εμφανίζεται κατάλληλο μήνυμα που επιβεβαιώνει ότι η είσοδος είναι σωστή.

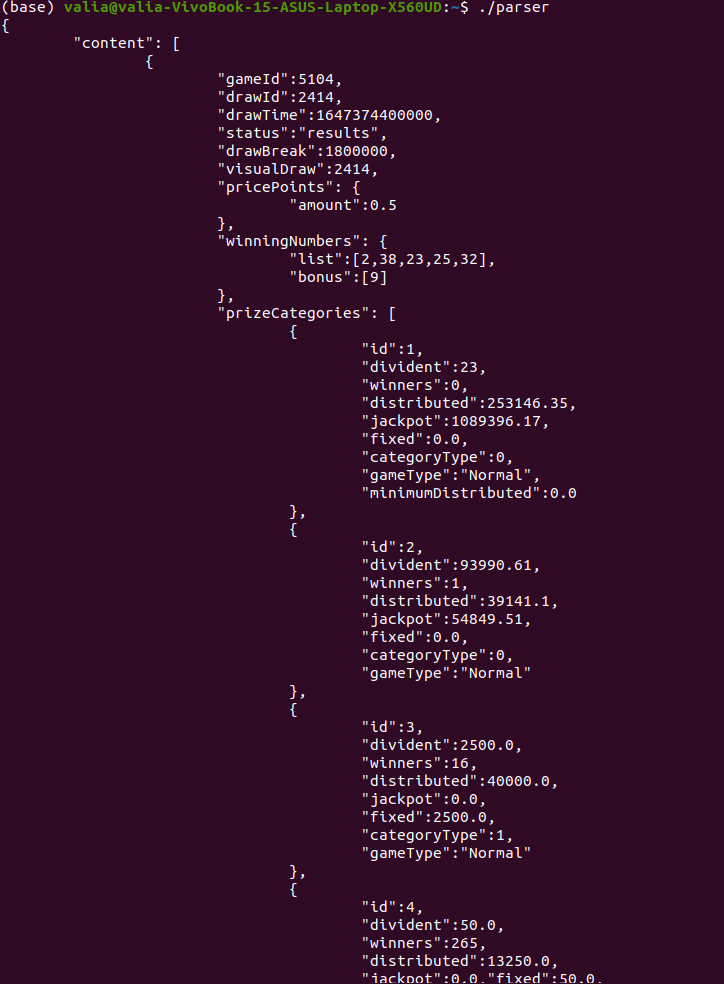
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Λανθασμένη Σειρά**

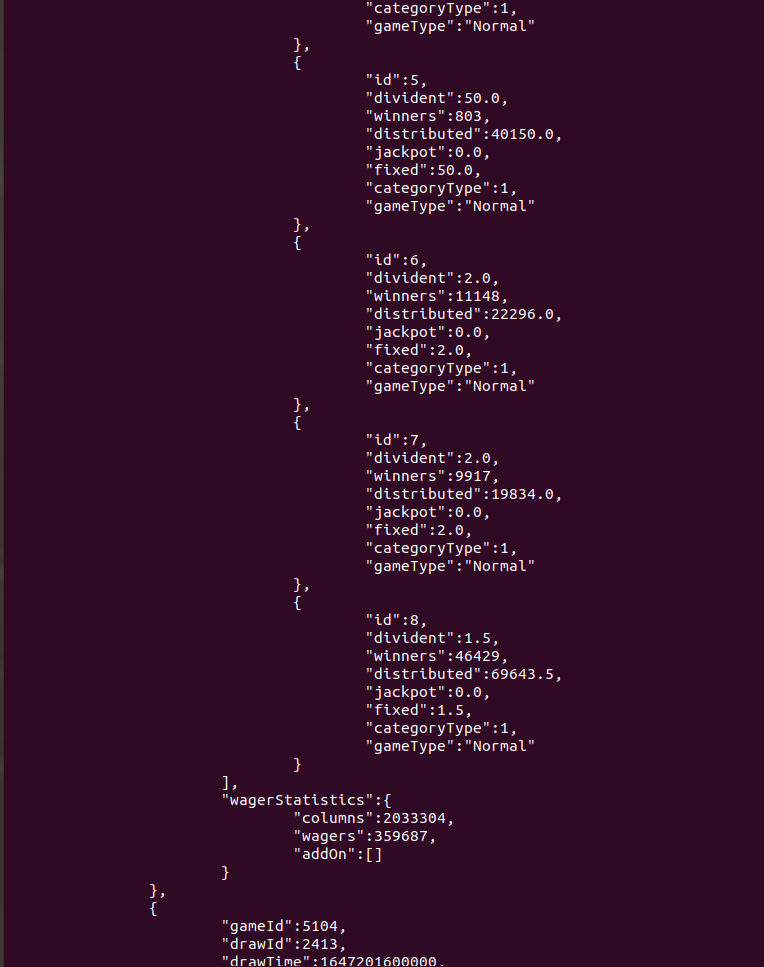


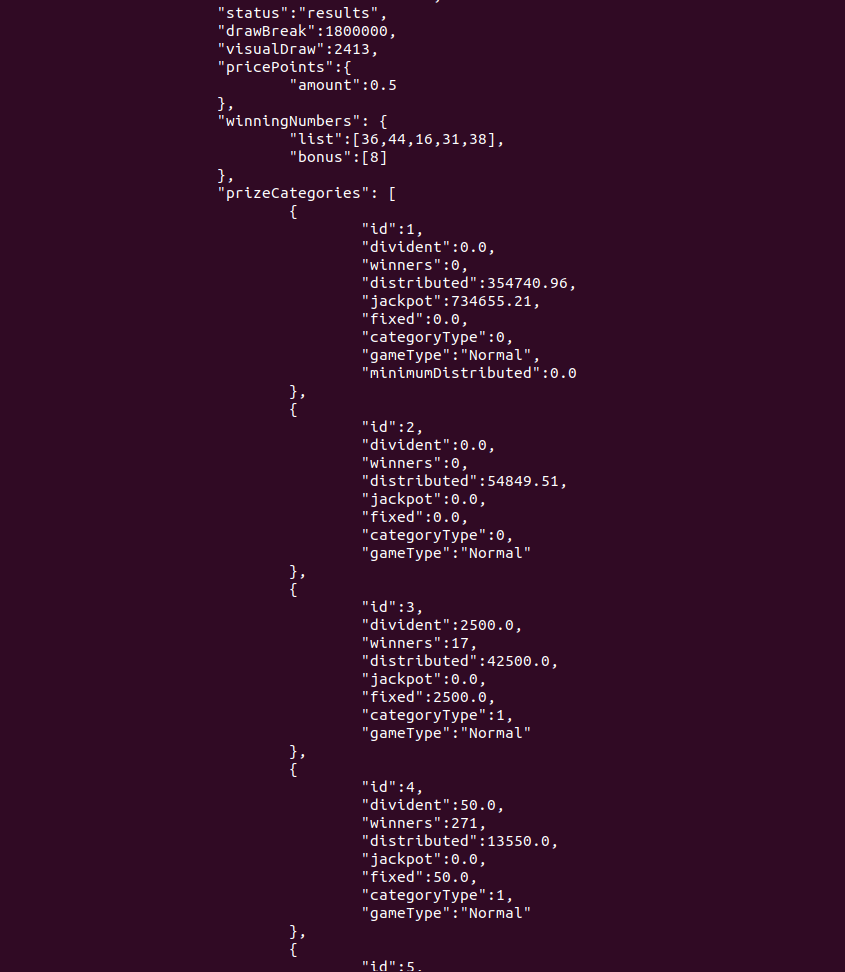


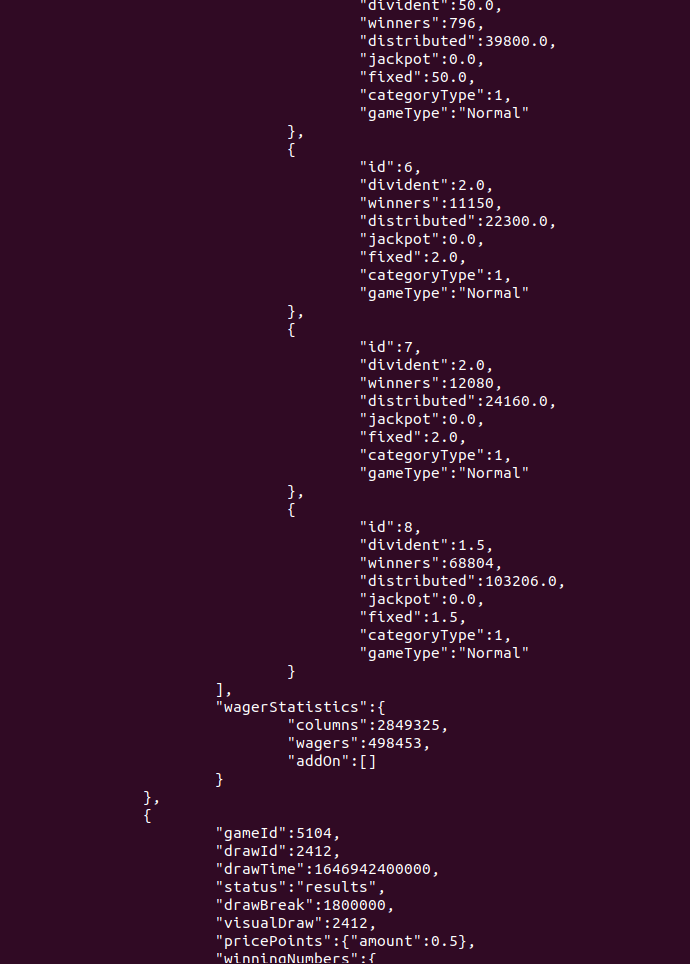
Παρατηρούμε πως αν αλλάξουμε τη σειρά των πεδίων drawTime και status, εμφανίζεται μήνυμα πως υπάρχει σφάλμα στη γραμμή 6. Αυτό συμβαίνει επειδή, ακόμη κι αν τεχνικά υπάρχει σφάλμα και στη γραμμή 7, το πρώτο σφάλμα εντοπίστηκε στην γραμμή 6 (στο πεδίο drawTime) και εκεί σταματάει η εκτέλεση.

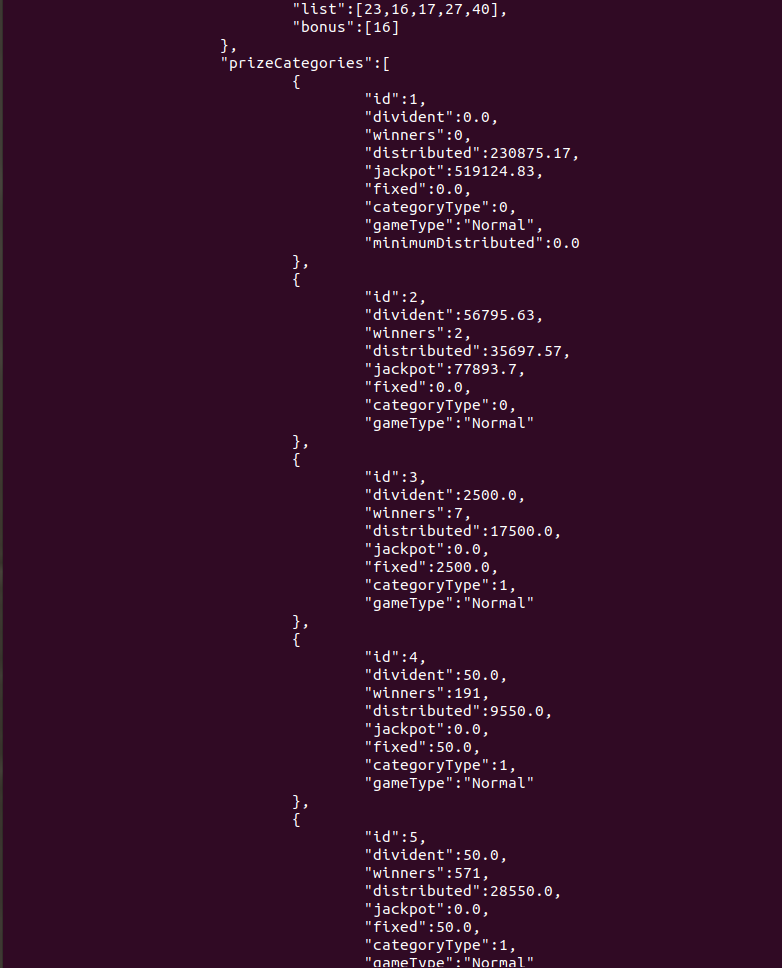
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Syntax error**

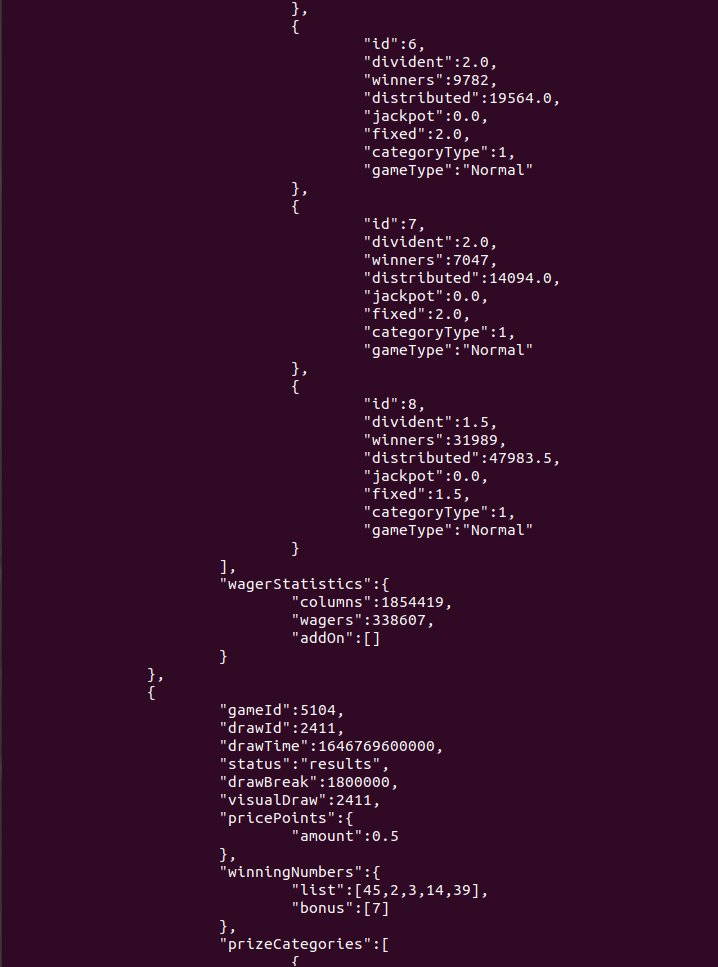


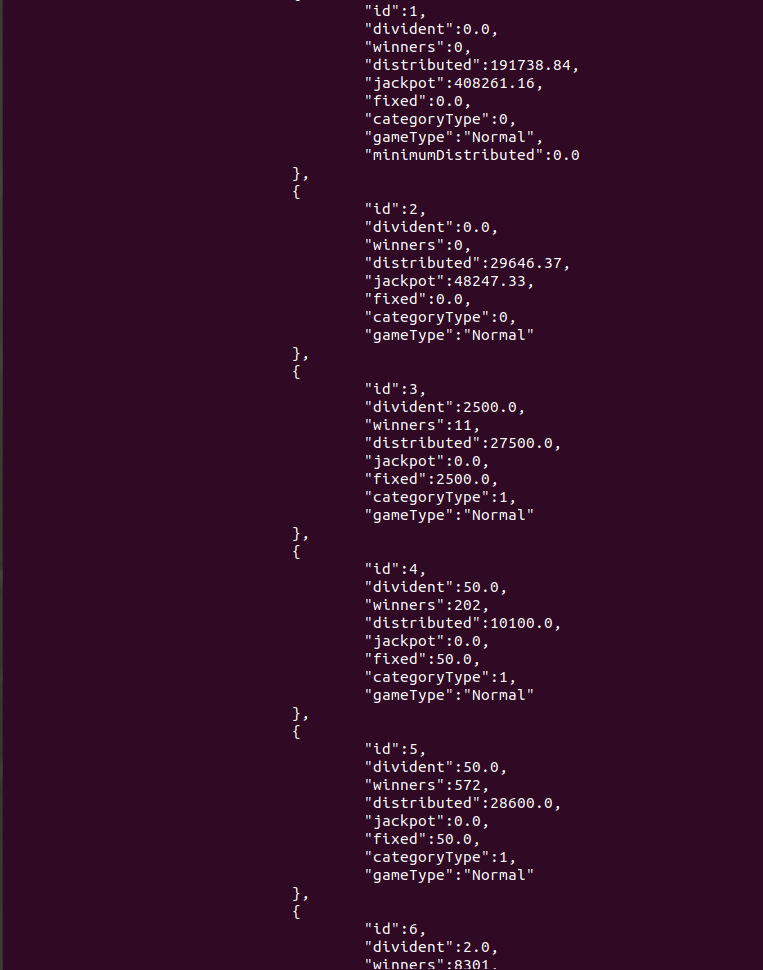








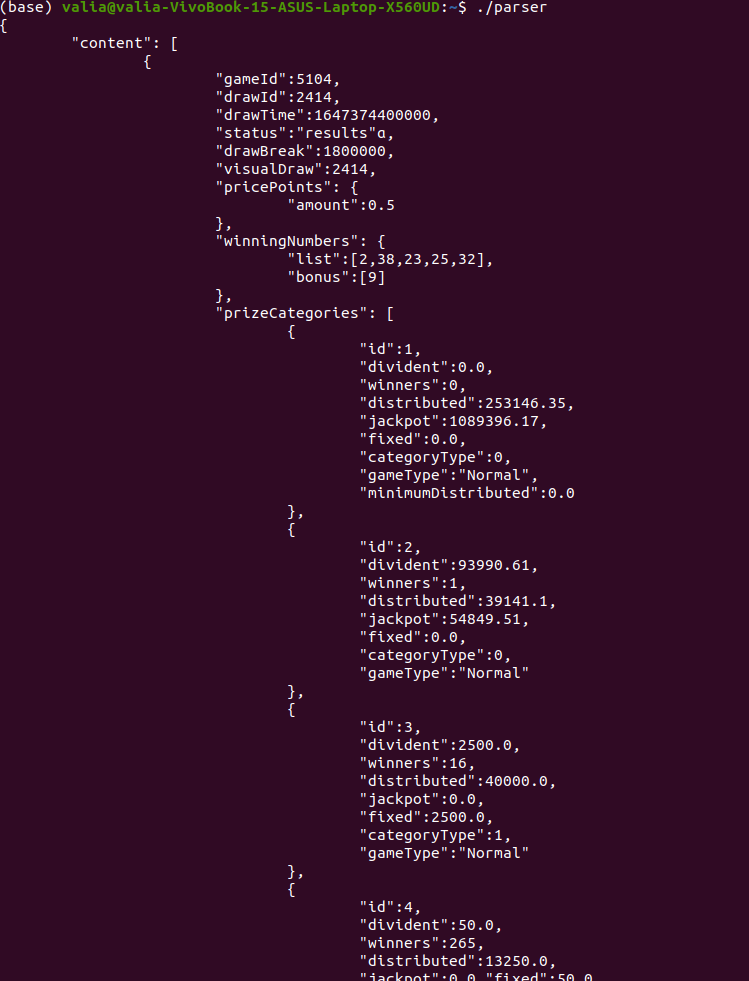


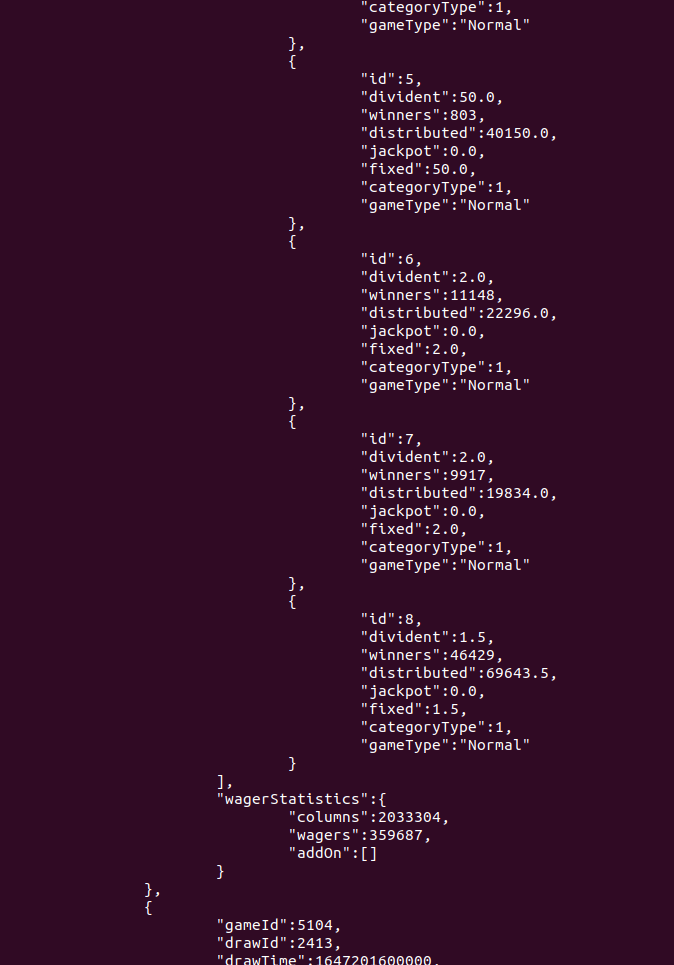


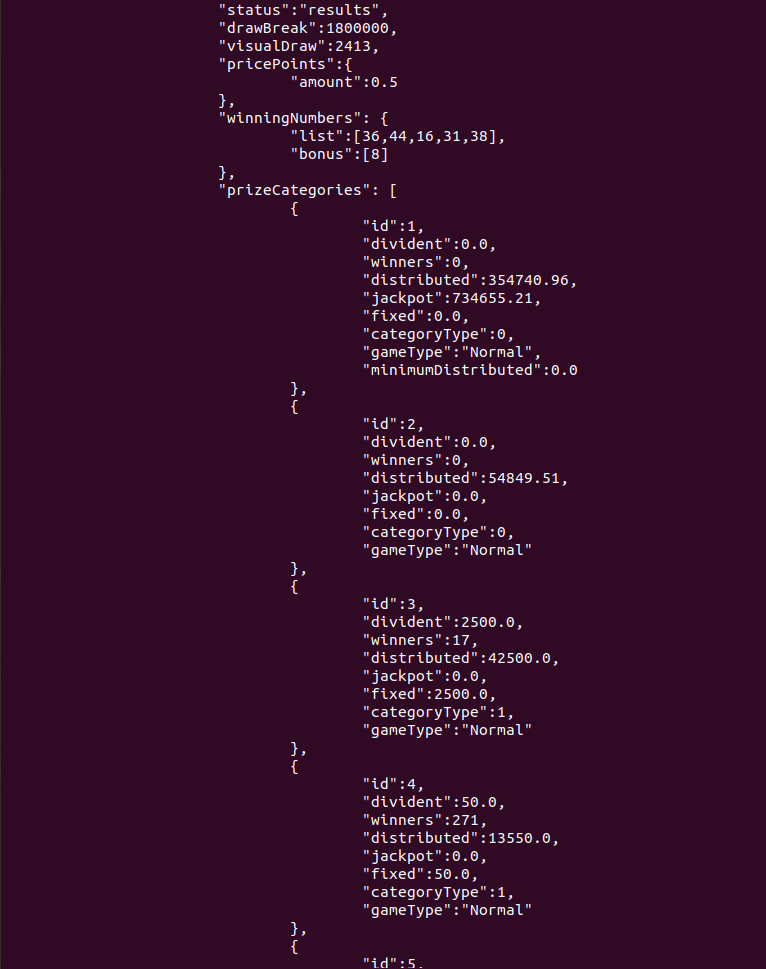


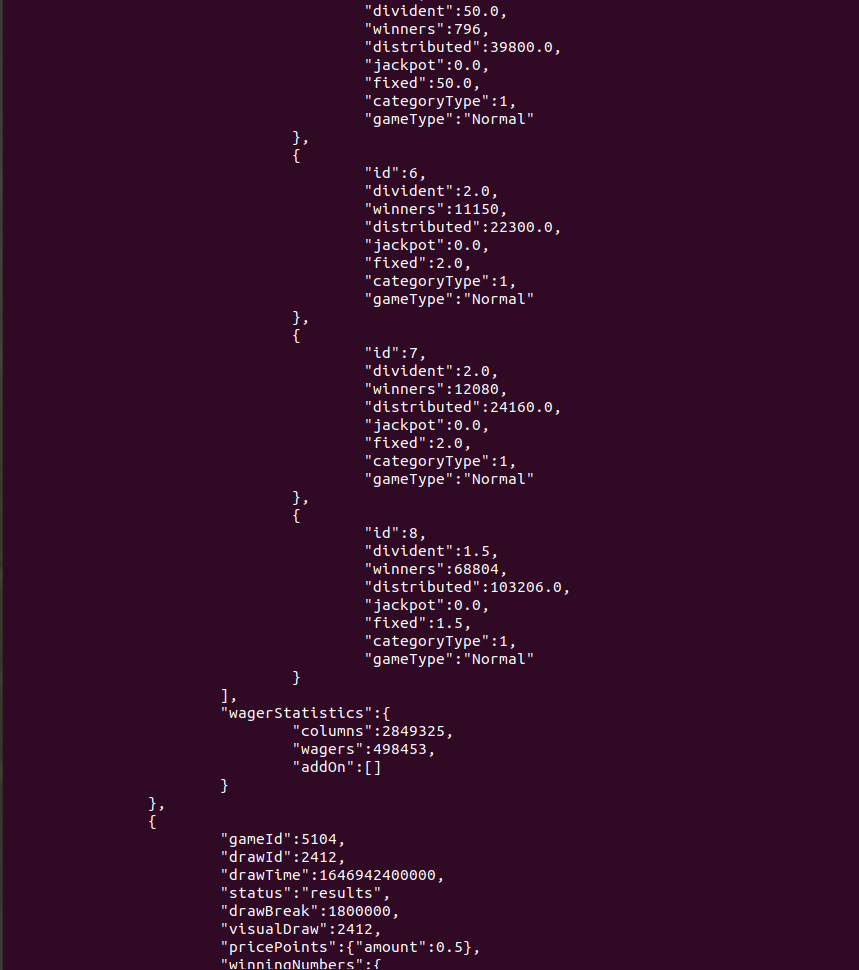
Παρατηρούμε ότι το σφάλμα προέκυψε στη γραμμή 20, όπου είναι το πεδίο divident. Το συγκεκριμένο πεδίο λαμβάνει πραγματικές τιμές, ενώ στη γραμμή 20 τοποθετήσαμε τον ακέραιο αριθμό 23. Ως αποτέλεσμα εμφανίζεται το αντίστοιχο μήνυμα για το σφάλμα syntax.

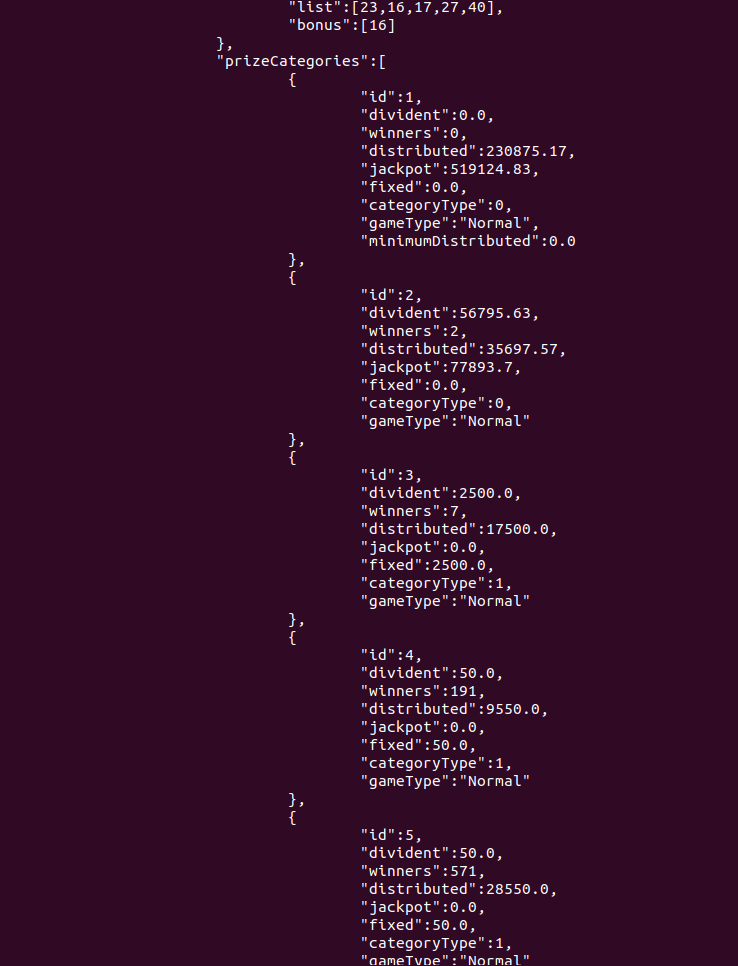
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : undefined symbol**

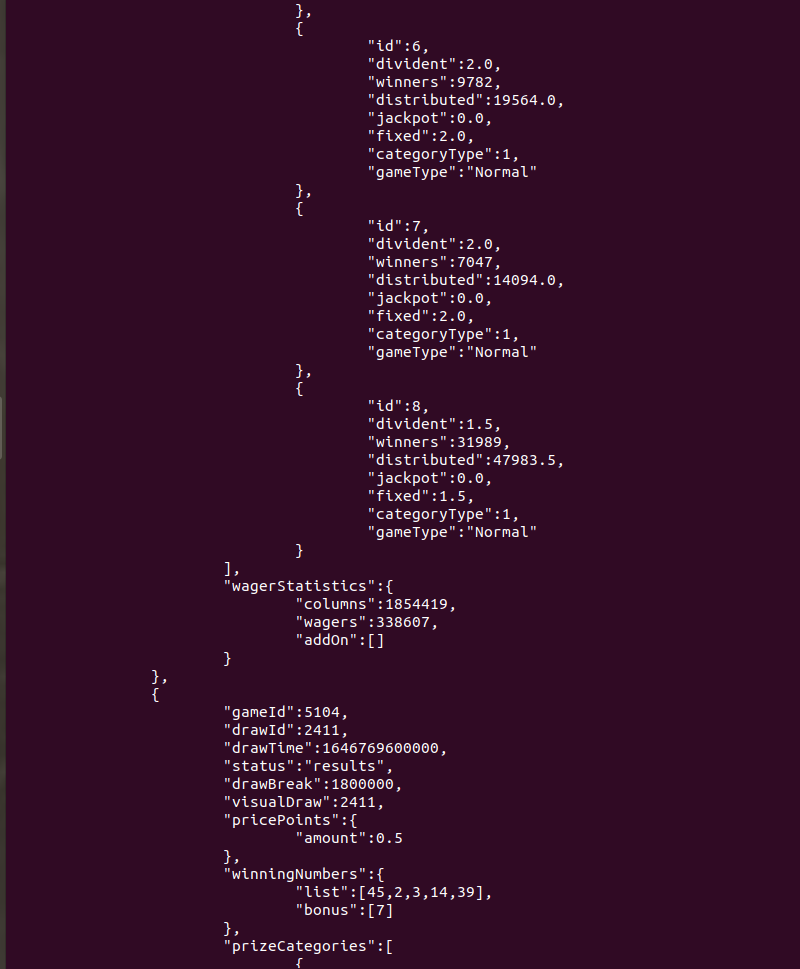


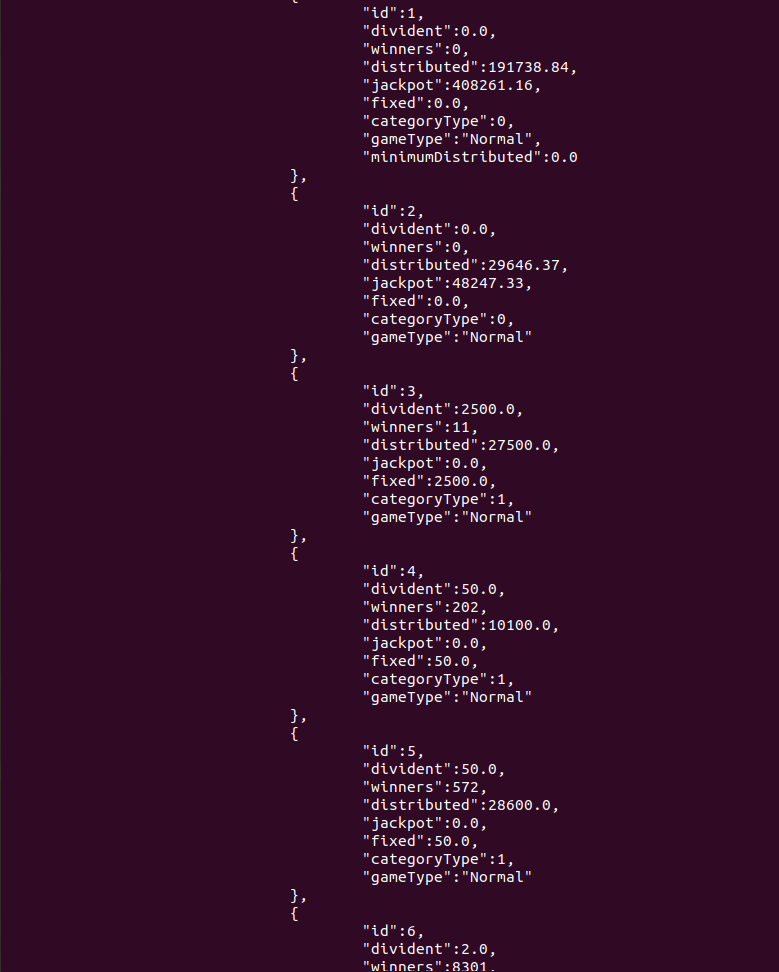














Στη γραμμή 7 της εισόδου βλέπουμε πως υπάρχει ένα ελληνικό “α” μετά το string που αναμένεται. Στη περίπτωση αυτή, προκύπτει σφάλμα καθώς το γράμμα δεν προβλέπεται ως επιλογή στο lex αρχείο που έχουμε. Συνεπώς, εκτυπώνεται αντίστοιχο μήνυμα σφάλματος

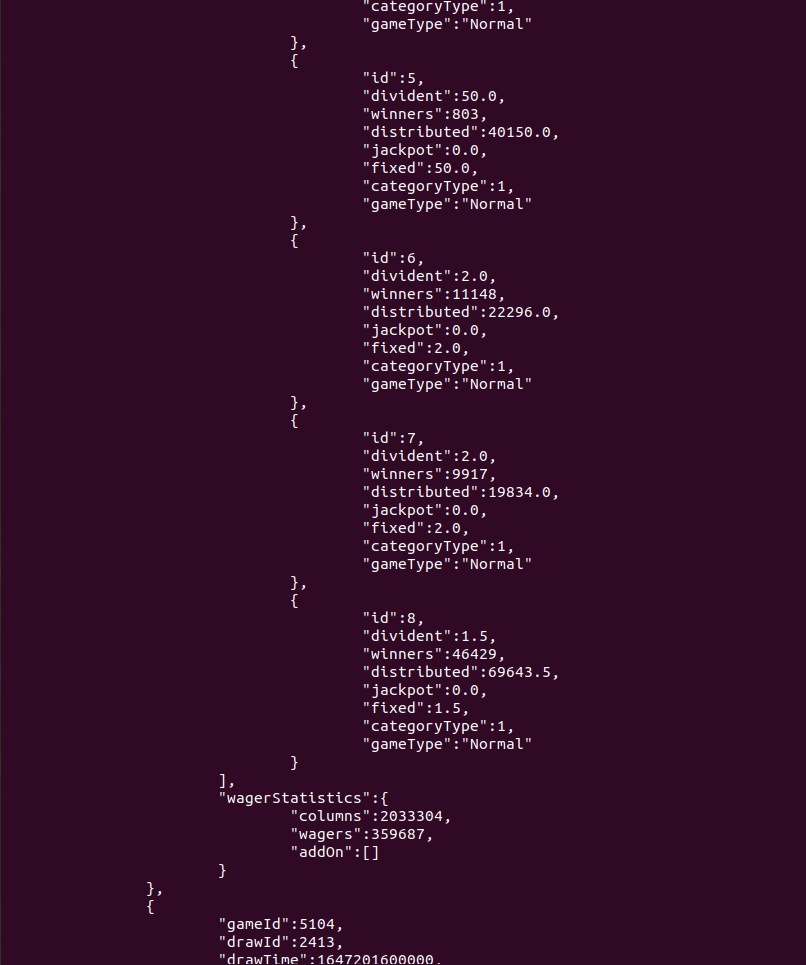
**Ερώτημα 3 :**

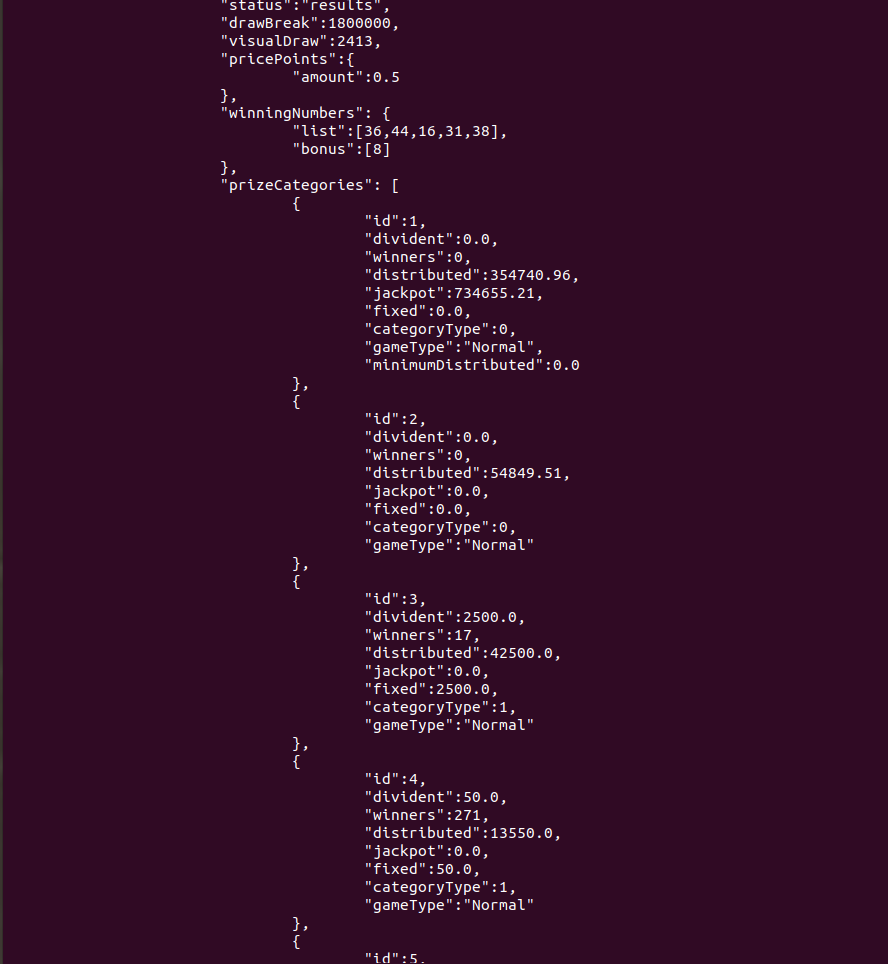
**H επιτυχημένη εκτέλεση του ερωτήματος 3 για το αρχείο last\_result εμφανίζει την ίδια έξοδο με την εκτέλεση των ερωτημάτων 1 και 2.**

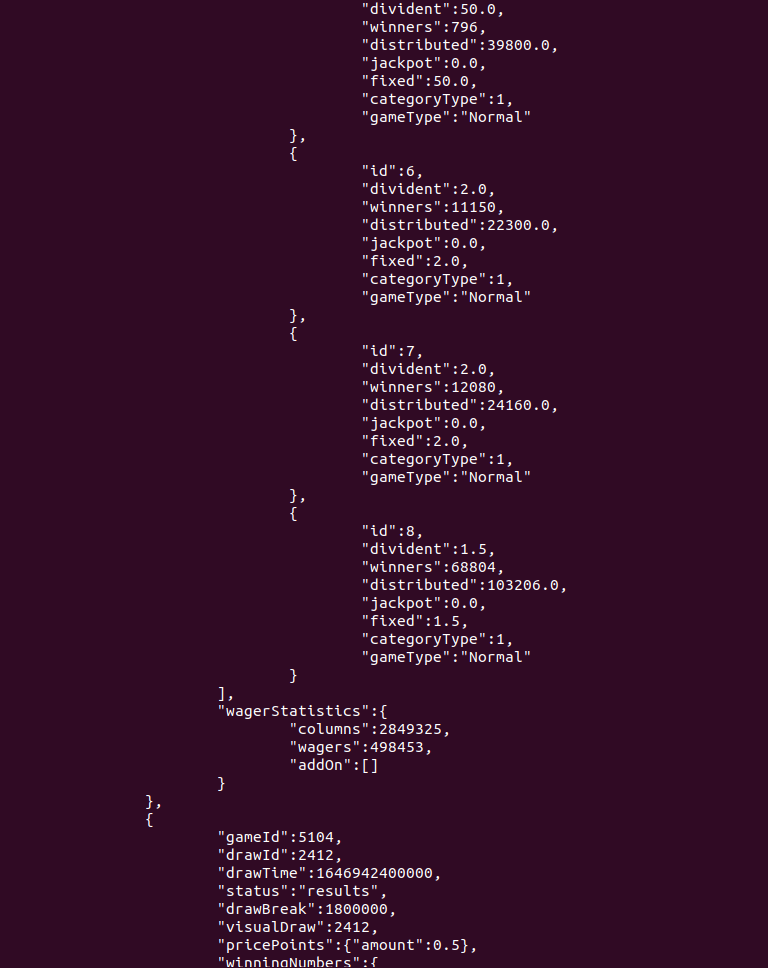
**H επιτυχημένη εκτέλεση του ερωτήματος 3 για το αρχείο range\_result εμφανίζει την ίδια έξοδο με την εκτέλεση του ερωτήματος 2.**

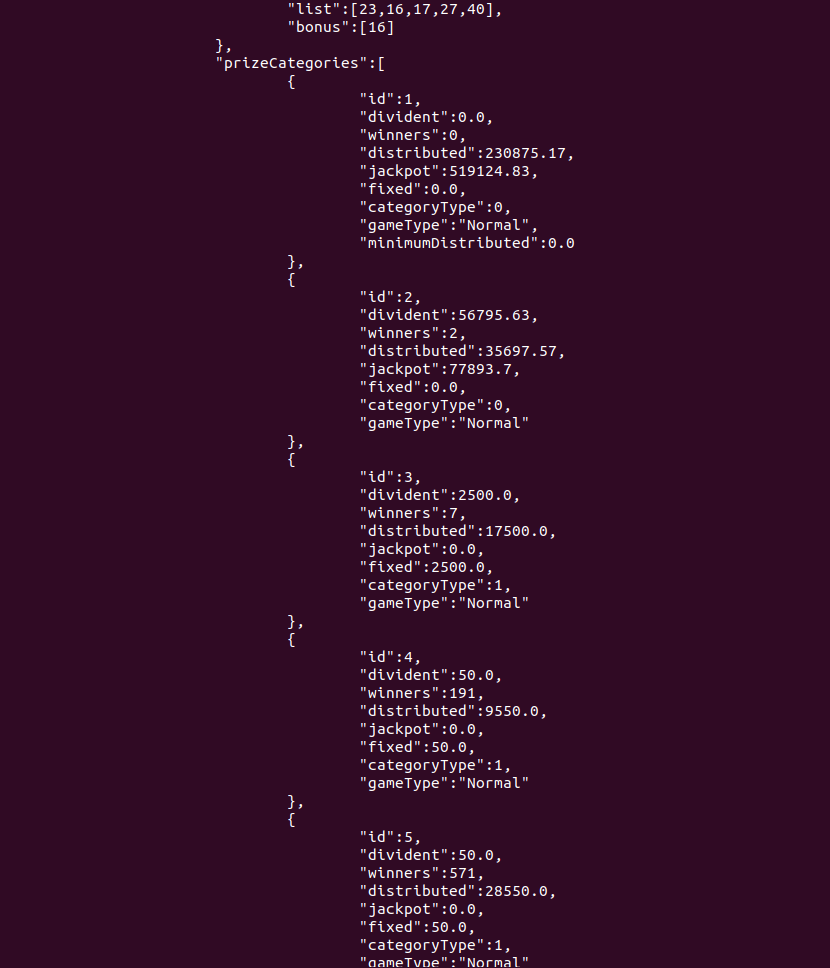
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Selection error [υποερώτημα (α)]**

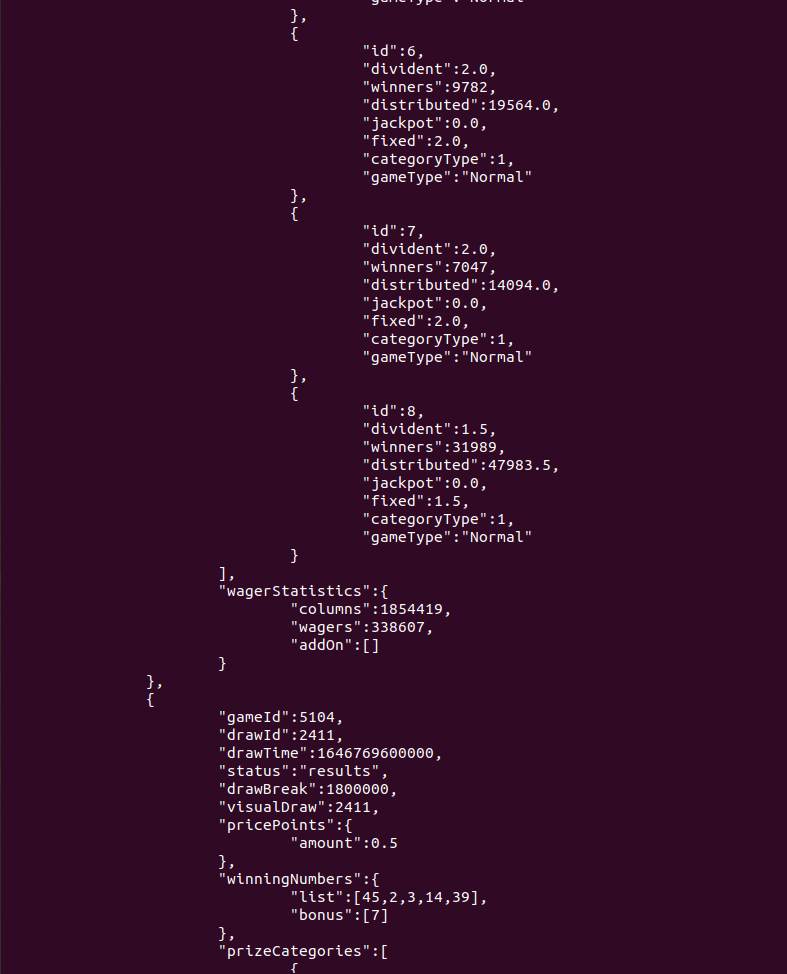


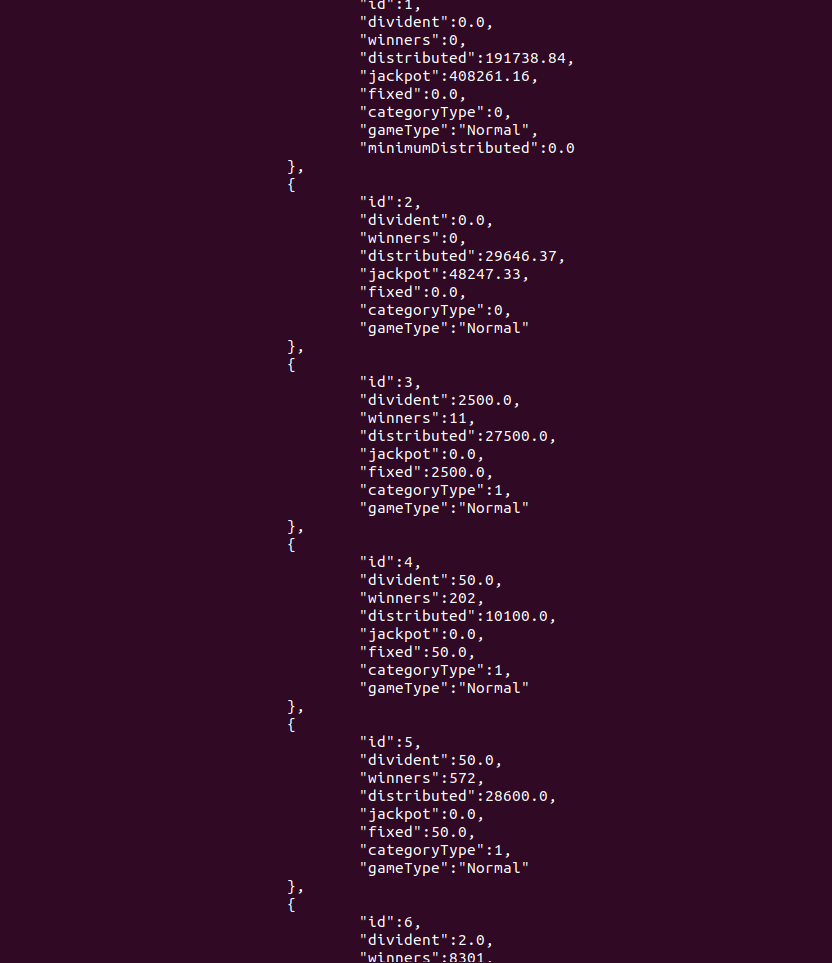








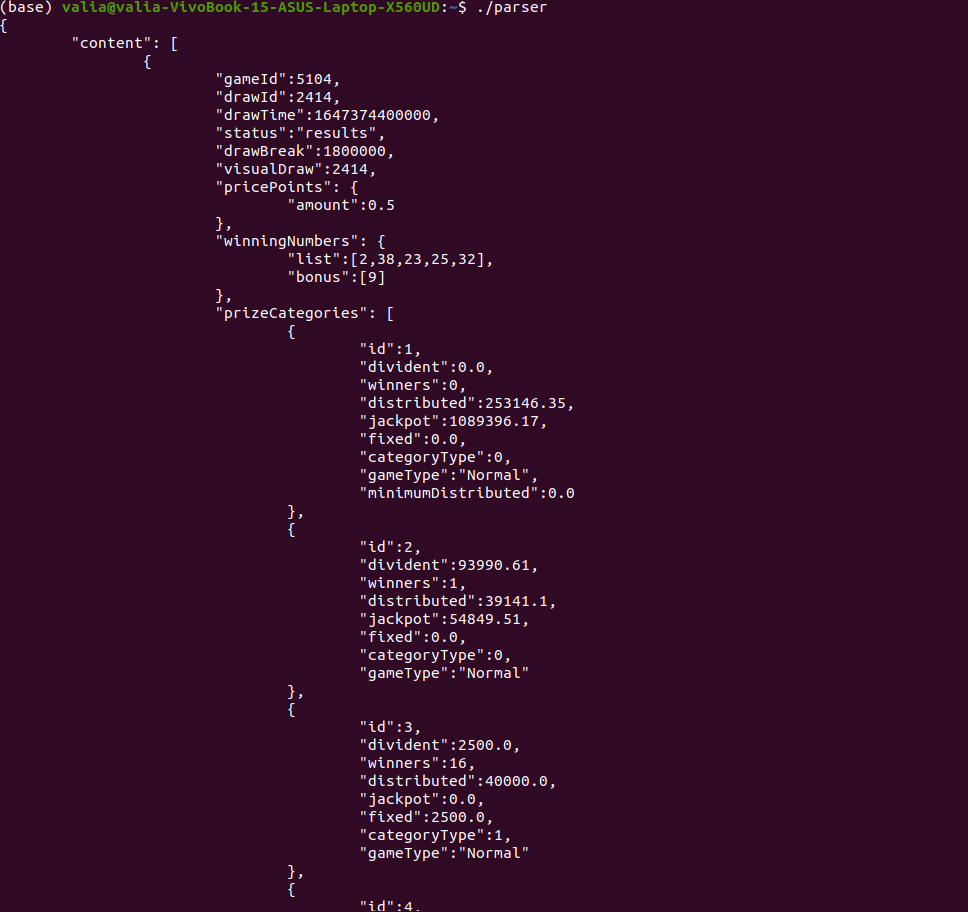


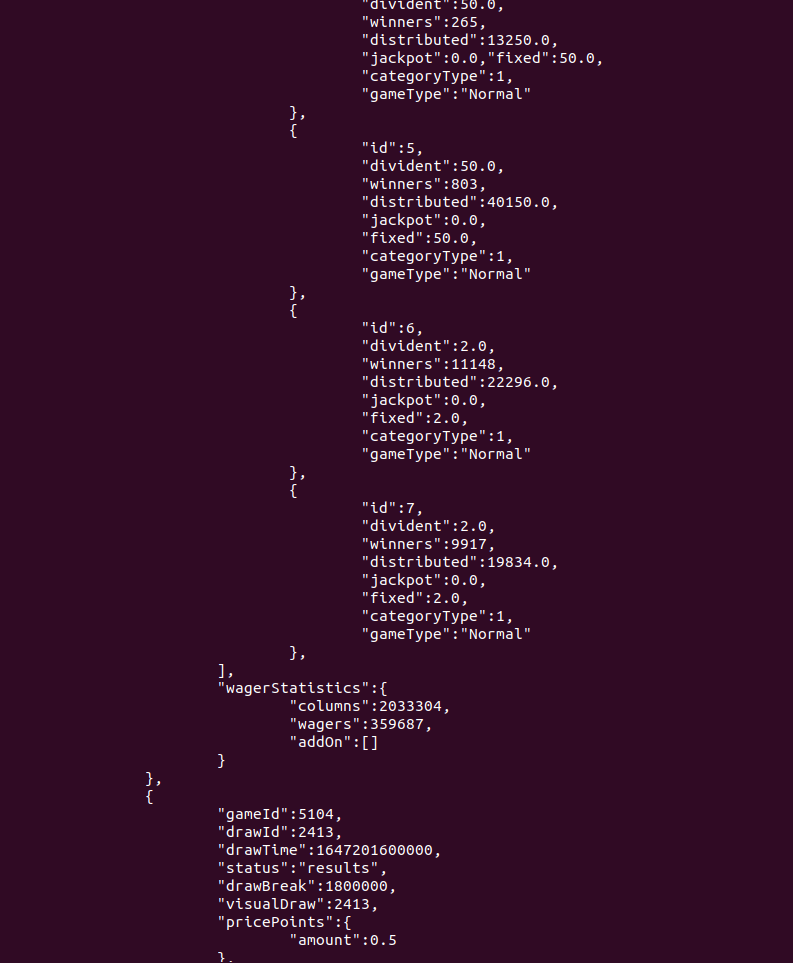


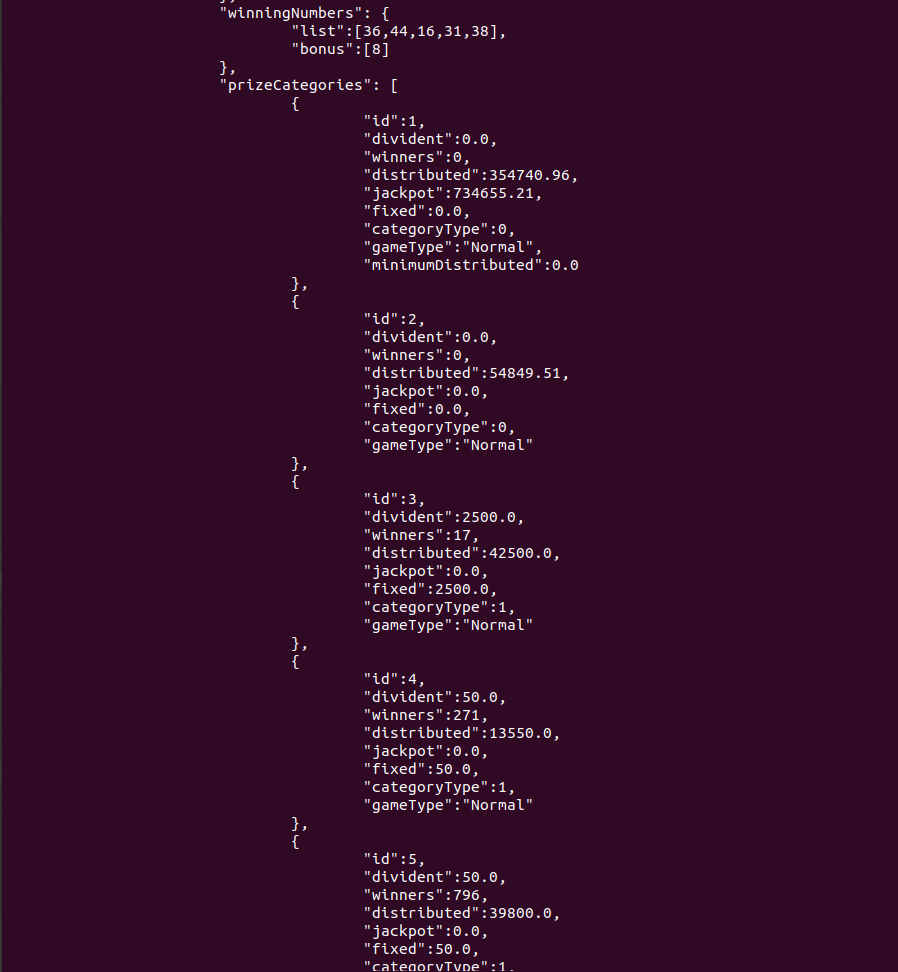


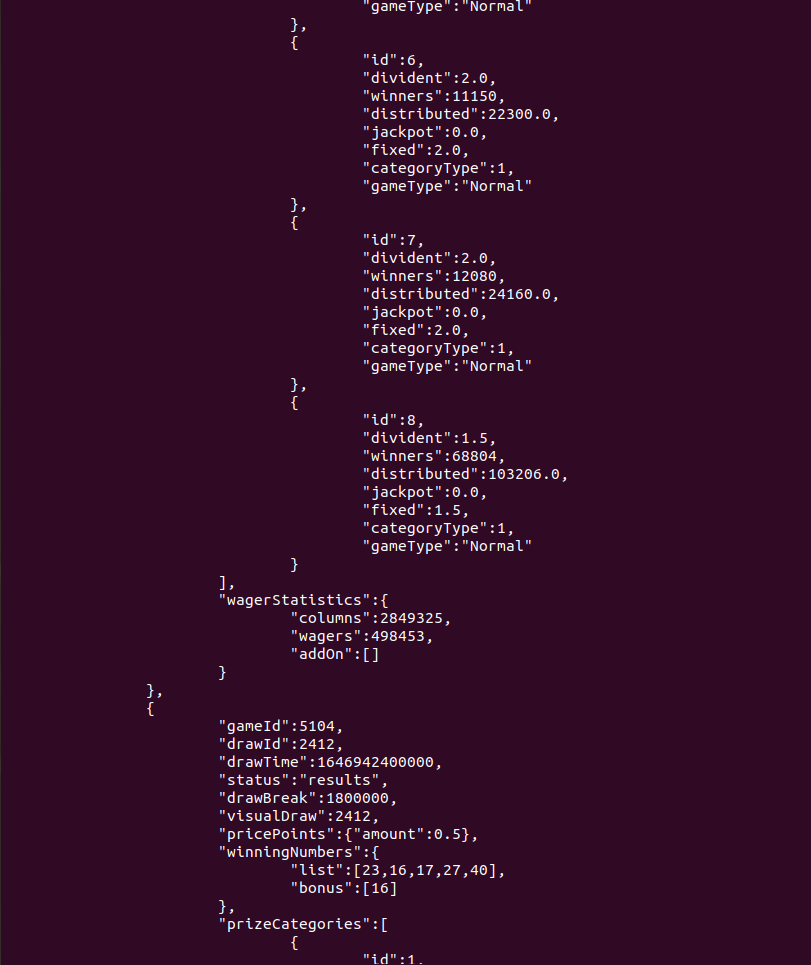
Στο υποερώτημα α ζητείται το πεδίο gameId να λαμβάνει συγκεκριμένες τιμές. Η τιμή 22 δεν υπάρχει στον πίνακα και ως εκ τούτου δεν υπάρχει και στους περιορισμούς που θέσαμε. Έτσι, δημιουργείται το σφάλμα wrong selection και εμφανίζεται το αντίστοιχο μήνυμα.

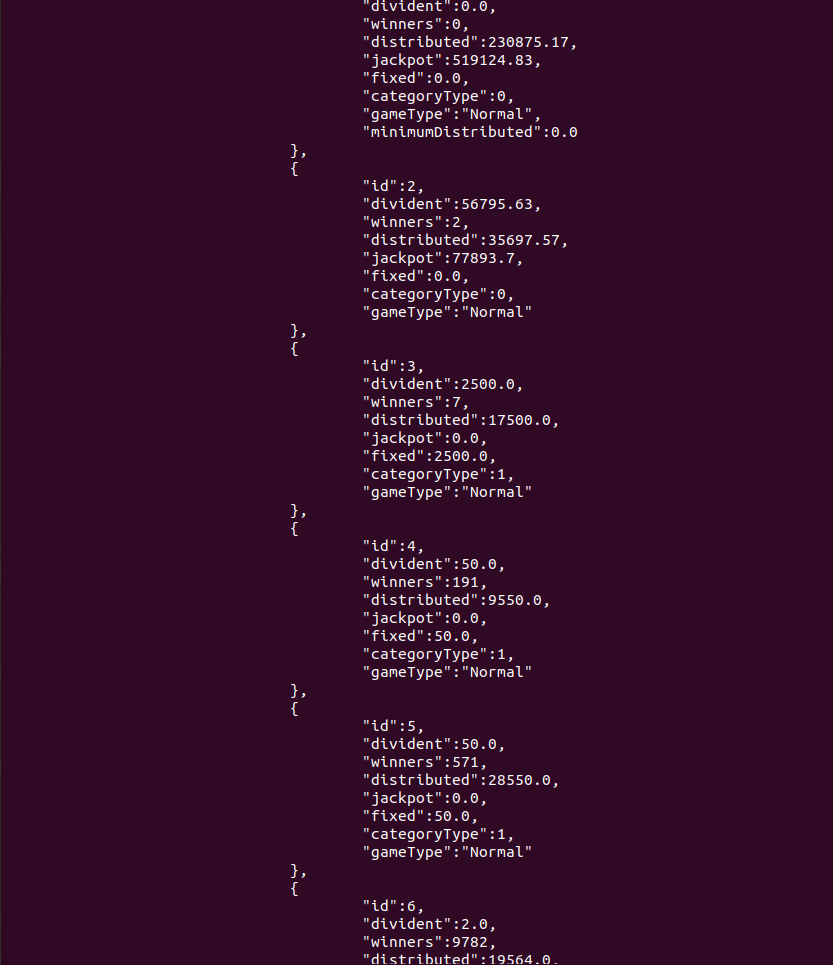
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Syntax error [υποερώτημα (b)]**

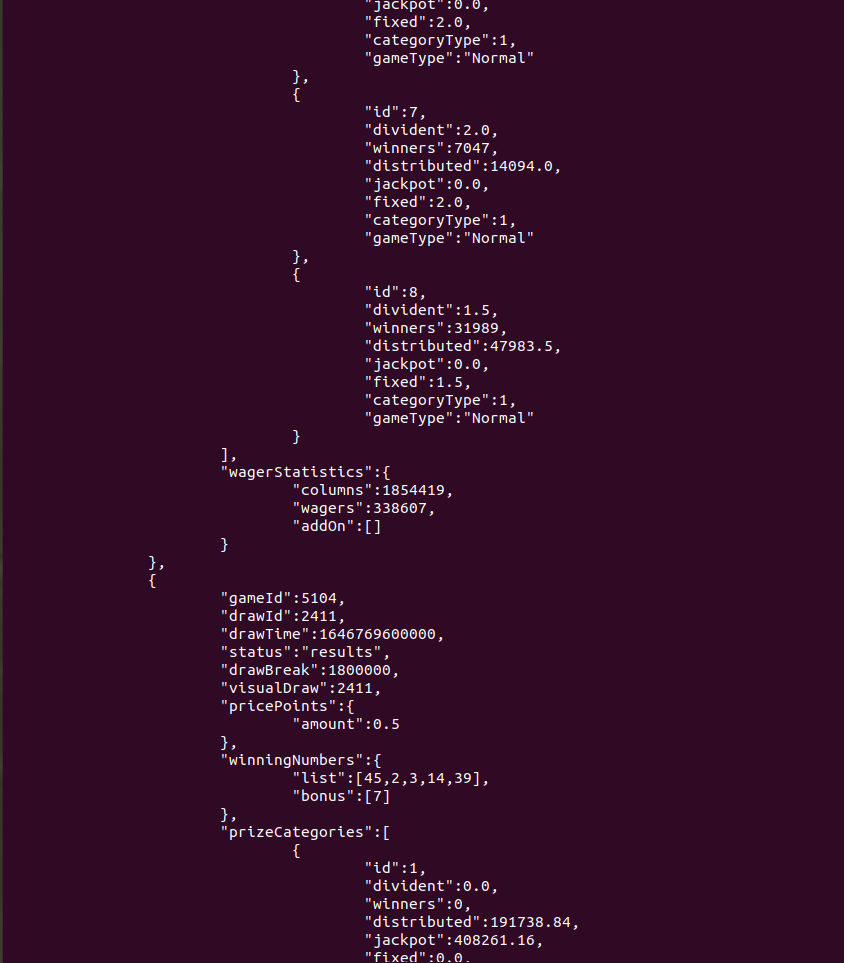




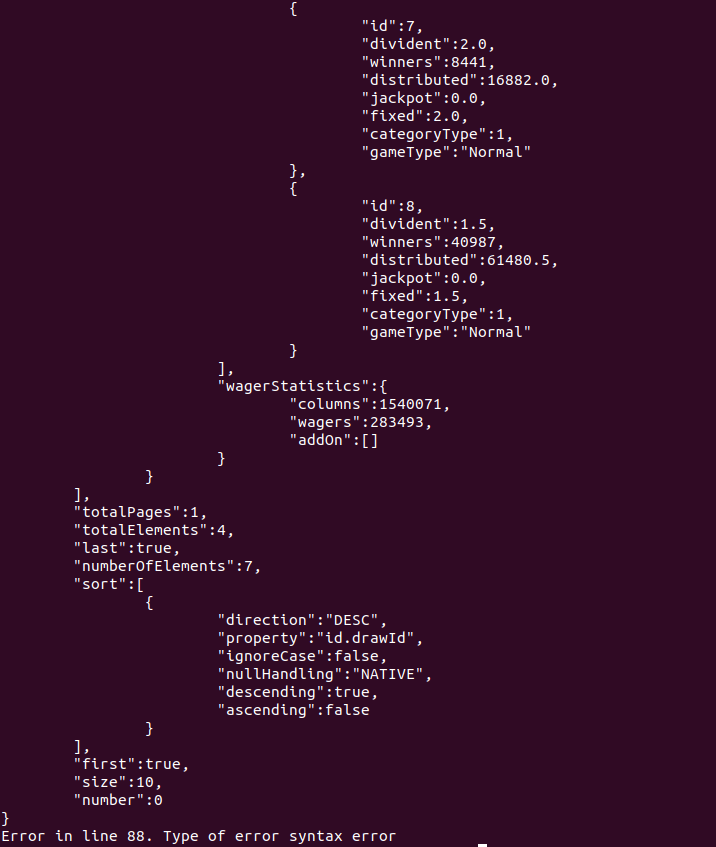






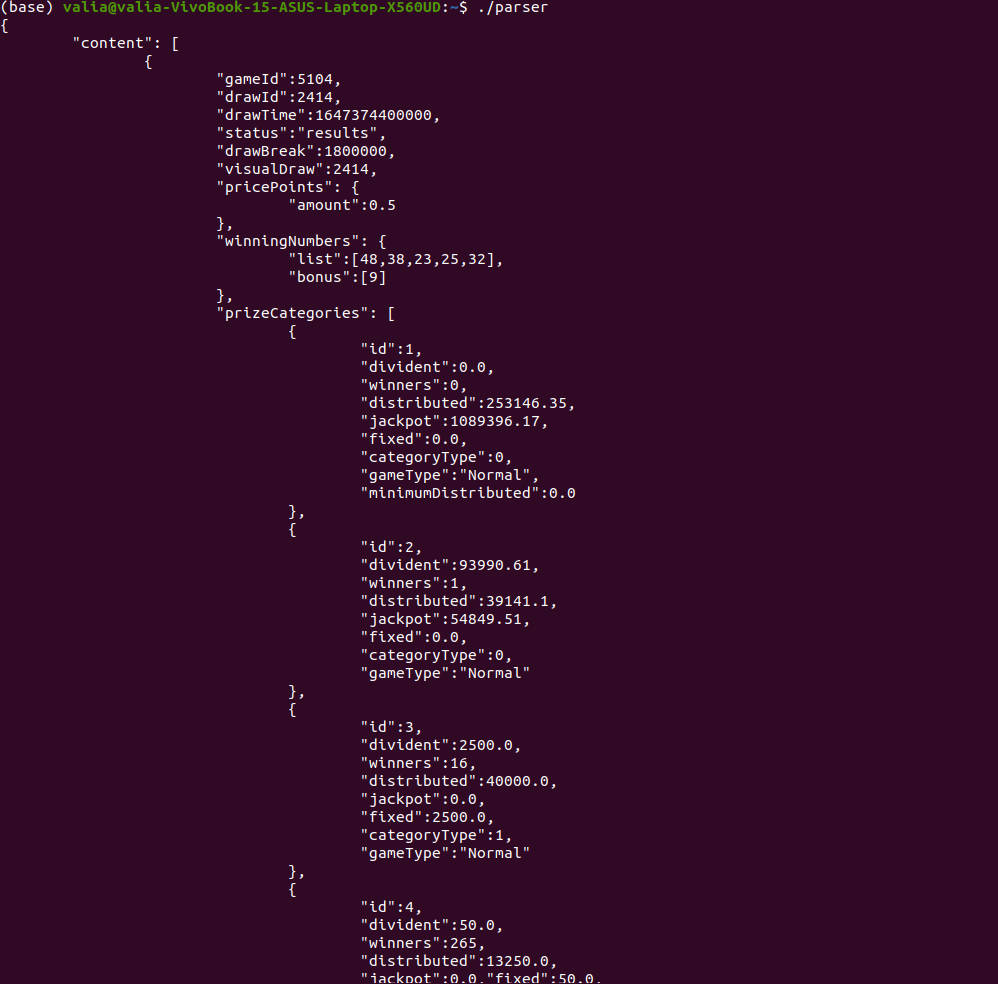


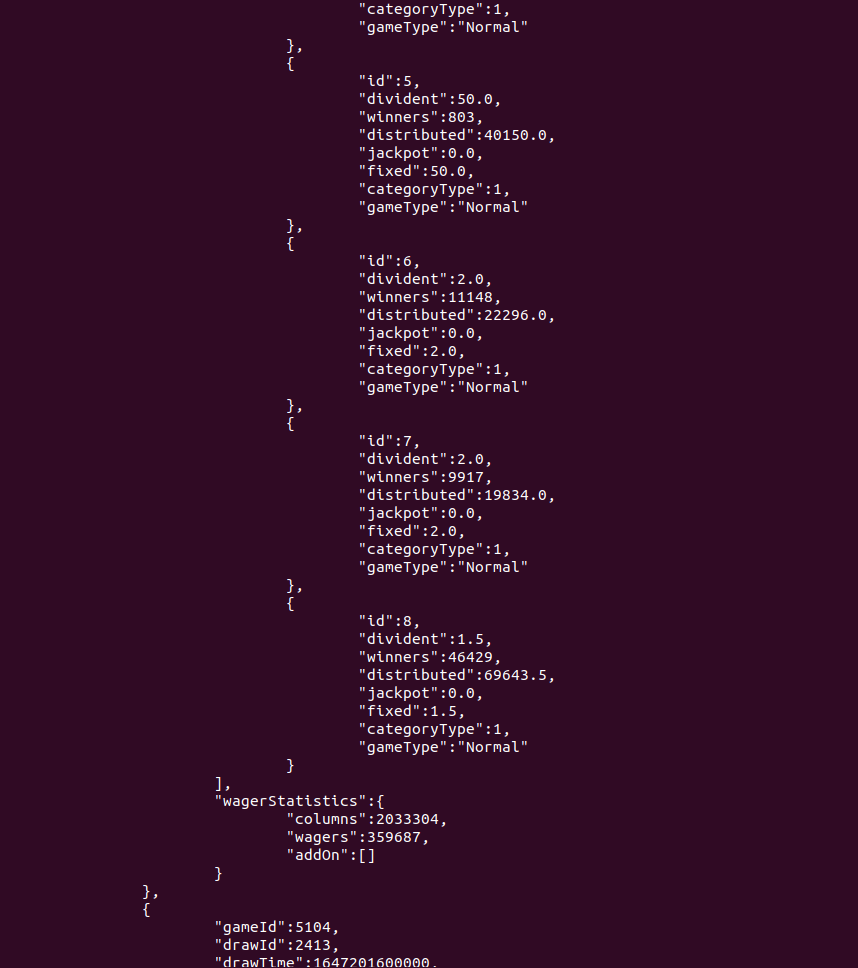


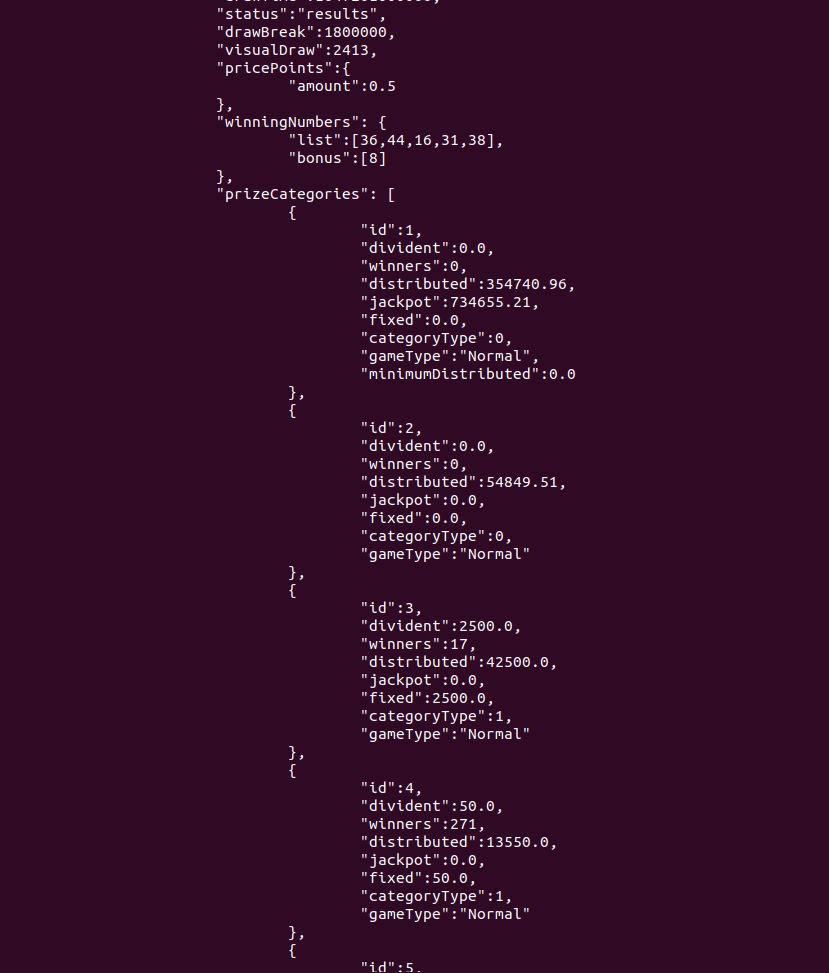


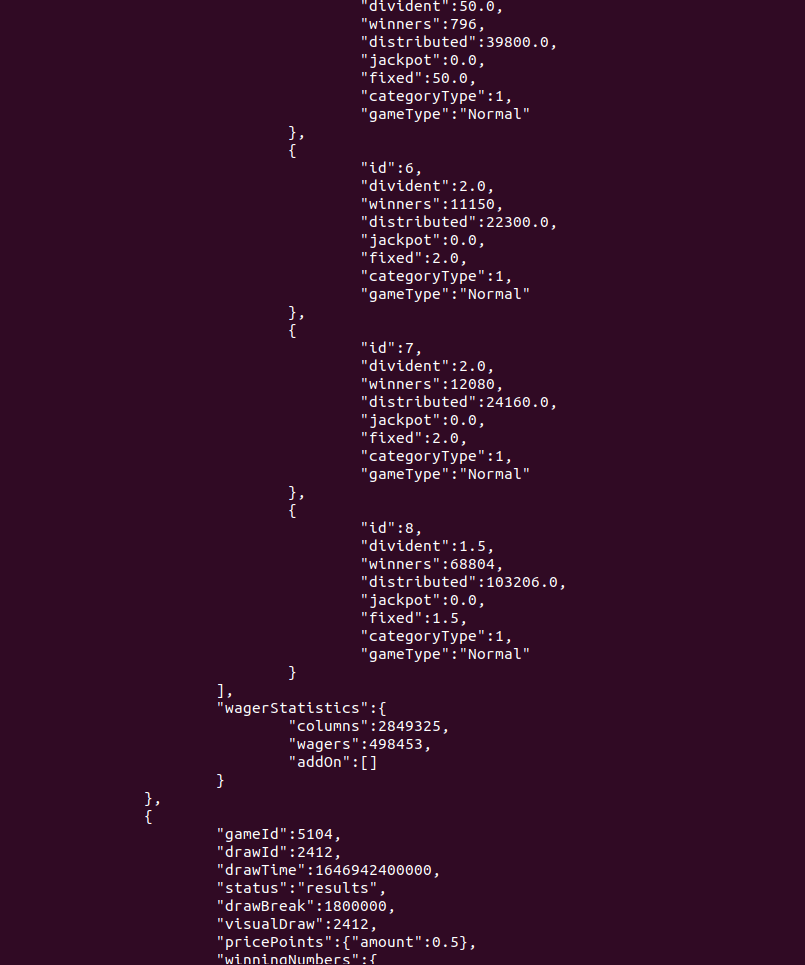
Παρατηρούμε πως στη γραμμή 88 εμφανίζεται το αντίστοιχο μήνυμα σφάλματος, καθώς το πεδίο prizeCategories περιέχει 7 εμφωλευμένα **json** αντικείμενα και όχι 8, όπως όλα τα υπόλοιπα πεδία prizeCategories.

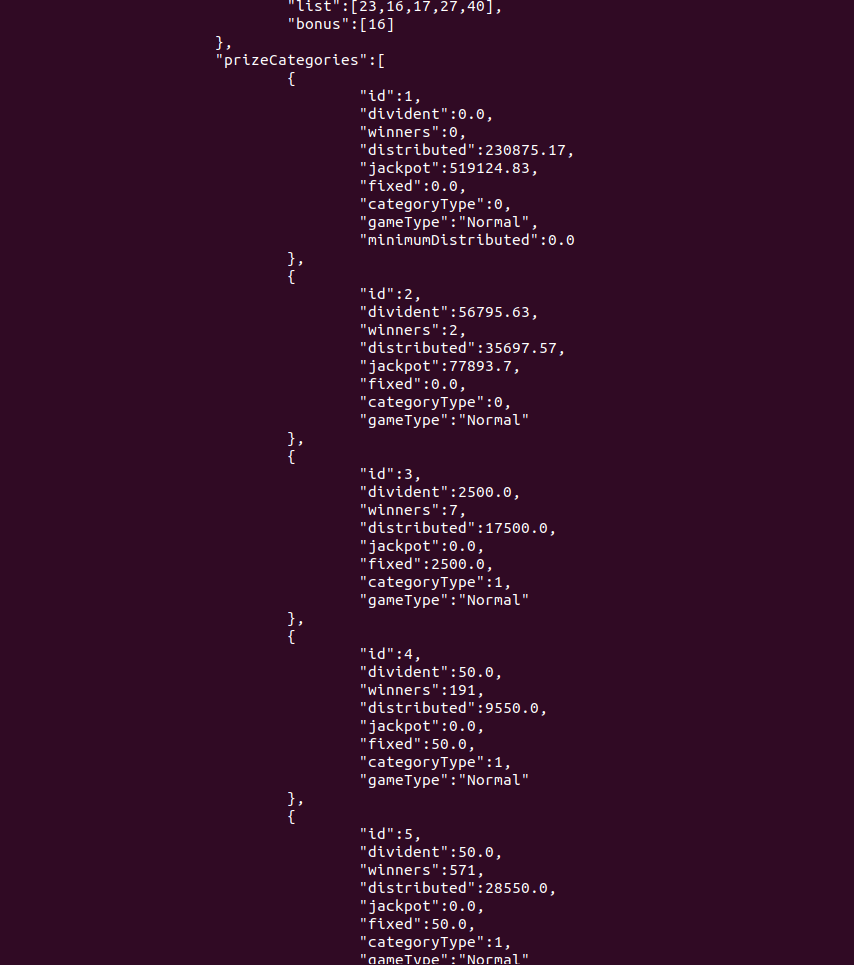
**Ανεπιτυχής εκτέλεση με τύπο σφάλματος : Value error [υποερώτημα (c)]**

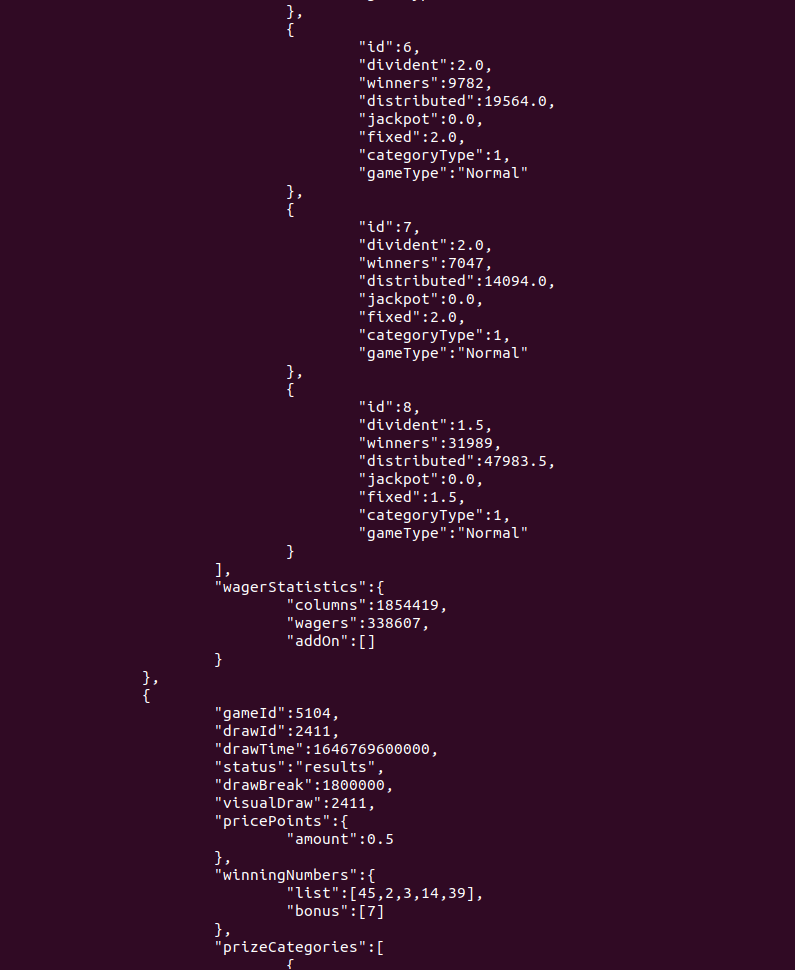


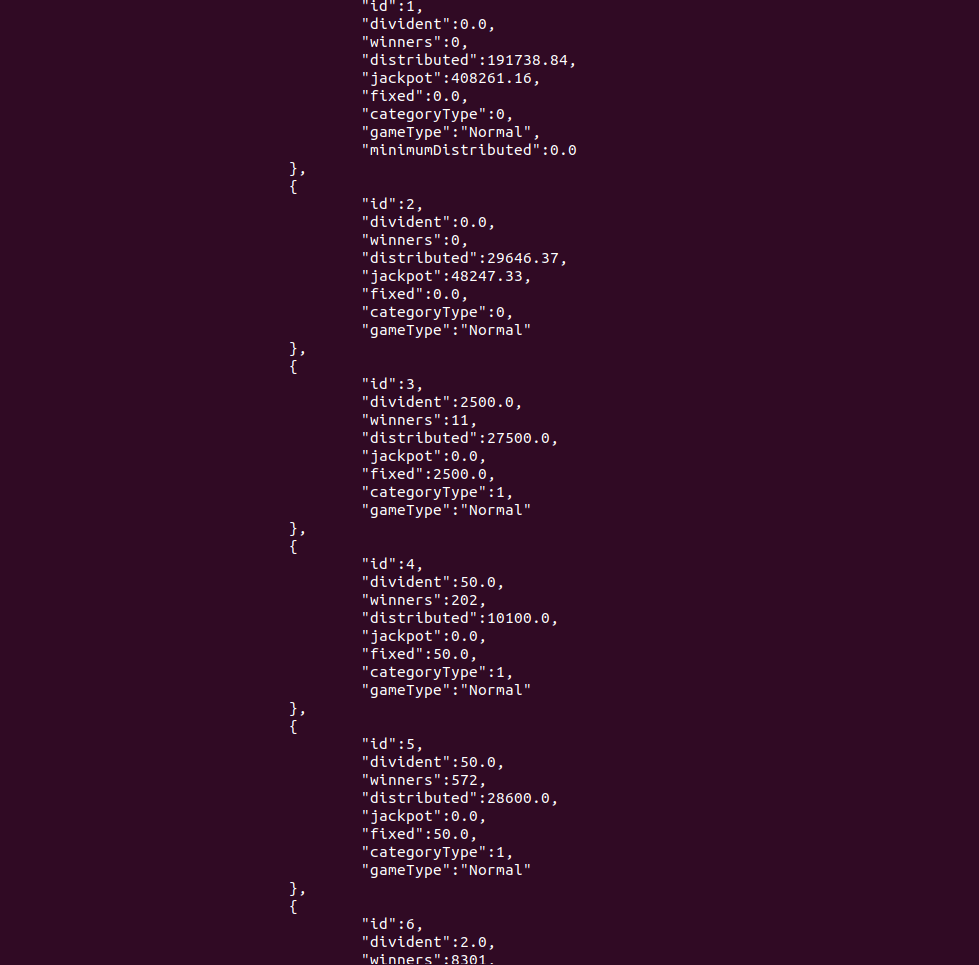














Στη γραμμή 14 βλέπουμε πως υπάρχει ο αριθμός 48, ενώ αν παρατηρήσουμε το αρχείο bison επιτρέπονται μόνο αριθμοί ανάμεσα στο 1 και το 45. Επομένως, εμφανίζεται το αντίστοιχο μήνυμα σφάλματος.

**Σχόλια και παραδοχές**

**Τα αρχεία FLEX**

Στα τρία αρχεία FLEX μπορούμε να δούμε ότι ορίζονται συγκεκριμένα σύμβολα. Αυτά είναι οι αγκύλες, η άνω-κάτω τελεία και το κόμμα. Τα συγκεκριμένα ορίζονται, έτσι ώστε να χρησιμοποιηθούν από το αρχείο BISON όταν διαβάζεται η είσοδος μορφής .json. Παράλληλα, μπορούμε να δούμε ότι υπάρχουν κανόνες για ακέραιους και πραγματικούς (θετικούς και αρνητικούς) αριθμούς. Παράλληλα, προστέθηκαν οι κανόνες για το new line, τα κενά και ο κανόνας της τελείας, ο οποίος σε περίπτωση σφάλματος επιστρέφει αντίστοιχο μήνυμα και τερματίζει το πρόγραμμα. Στο τρίτο αρχείο FLEX, δημιουργήθηκε ένας πρόσθετος κανόνας BOOL για να χρησιμοποιηθεί από τη BISON όταν δέχεται το αρχείο range\_result στην είσοδο. Τέλος, δημιουργήσαμε μία κανονική έκφραση WORD, που χρησιμοποιείται για την αναγνώριση των String. Το regex \"(\\.|[^"\\])\*\" , εξηγείται ως εξής :

Κάθε string περιέχει ένα εισαγωγικό [ ” ] στην αρχή, το οποίο ακολουθείται από μηδέν ή περισσότερους escaped χαρακτήρες που συμβολίζεται ως [ \\. ] ή από έναν χαρακτήρα που δεν είναι εισαγωγικό, ούτε κάθετος, δηλαδή [ ^”\\ ]. To σύμβολο [ ^ ] είναι ο συμβολισμός της άρνησης. Τέλος, υπάρχει ένα τερματικό εισαγωγικό και συνολικά τοποθετώντας τις καθέτους για ειδικούς κανόνες έχουμε \"(\\.|[^"\\])\*\".

Η εντολή yylval.ch=strdup(yytext); χρησιμοποιείται για να περάσει μία τιμή που είναι συνδεδεμένη με το αντίστοιχο token, από τον lexer στον parser. Λαμβάνουμε την τιμή ενός συγκεκριμένου token μέσω του yytext.

**Τα αρχεία BISON**

Στα αρχεία BISON, δημιουργήσαμε ένα union για τους τύπους δεδομένων char, int και float. Παράλληλα, μπορούμε να δούμε τον ορισμό των tokens που βρίσκονται στο αρχείο FLEX. Κάθε κανόνας BISON που αντιστοιχεί σε πεδίο του αρχείου .json συνοδεύεται από μία εντολή της μορφής {char target[]={'"',[**πεδίο**] ,'"','\0'};if(strcmp($1,target)!=0){yyerror(":wrong sequence");YYERROR;};}; , η οποία συγκρίνει το string της εισόδου και ελέγχει αν συμπίπτει με την επιθυμητή μορφή που προβλέπεται να έχει η είσοδος. Αν τα δύο string διαφέρουν, καλείται η YYERROR και εκτυπώνεται μήνυμα λάθους ακολουθίας χαρακτήρων. Στον κανόνα για το πεδίο ID, παρατηρούμε πως αν η τιμή στην είσοδο δεν είναι στο διάστημα 1 έως 8, εκτυπώνεται κατάλληλο μήνυμα και καλείται η YYERROR. Επιπρόσθετα, η χρήση των ειδικών συμβόλων είναι εμφανής στους κανόνες της BISON, καθώς η δομή των εκφράσεων συνήθως αποτελείται από εκφράσεις των tokens που χρησιμοποιούνται για να συμβολίσουν τις αγκύλες, την άνω-κάτω τελεία και το κόμμα.

**Τα BNF των τριών ερωτημάτων**

Στα BNF μπορούμε να δούμε πως υπάρχουν εκφράσεις για όλα τα πεδία που προβλέπεται να περιμένουμε στην είσοδο. Ωστόσο, υπάρχουν πρόσθετες εκφράσεις οι οποίες υπάρχουν για να διευκολύνουν την επεξεργασία του αρχείου. Οι εκφράσεις **C, FAA, FAD, AT, AAA, ADA** χρησιμοποιούνται για τα ειδικά σύμβολα που υπάρχουν στη δομή του αρχείου. Εκφράσεις όπως οι **PPW, WNW, prizeCatW, WSW, listW, bonusW** αποτελούνται από ειδικά σύμβολα και την έκφραση word, το οποίο σημαίνει πως χρησιμοποιούνται από το αρχείο **BISON** για να αναγνωρίζει τα ονόματα των πεδίων και τα ειδικά σύμβολα που τα ακολουθούν. Παράλληλα, για να μπορεί η **BISON** να αναγνωρίσει πίνακες με πολλαπλές τιμές χρησιμοποιήθηκε αναδρομή στις εκφράσεις, γεγονός το οποίο είναι εμφανές στα πεδία **larray, addOnEXP, contentAR.** Στα ερωτήματα δύο και τρία, η είσοδος μπορεί να έχει δύο μορφές, επομένως η έκφραση start περιέχει δύο επιλογές. Η πρώτη συμπίπτει με του ερωτήματος ένα και δέχεται αρχεία της μορφής **last\_result.json**, ενώ η επιλογή **startR** δέχεται αρχεία μορφής **range\_result.json**. H δεύτερη επιλογή περιέχει και πάλι όλα τα αναγκαία πεδία, έτσι ώστε να αναγνωρίζεται η ορθή μορφή του αρχείου. Σε αυτή τη περίπτωση, μπορούμε να δούμε πως ακολουθείται η ίδια δομή στις εκφράσεις, είτε αυτές ελέγχουν το όνομα του πεδίου, είτε λαμβάνουν κάποια τιμή (**word, posI, negI, posR, negR, bool**).