# Αρχές Γλωσσών Προγραμματισμού & Μεταφραστών
# Τμήμα Μηχανικών Η/Υ & Πληροφορικής
# Πανεπιστήμιο Πατρών
# Εαρινό Εξάμηνο 2022
# Διδάσκοντες: Ι. Γαροφαλάκης, Σ. Σιούτας, Π. Χατζηδούκας


## ΜΕΛΗ ΟΜΑΔΑΣ:

- ΒΛΑΣΙΟΣ ΠΑΝΑΓΙΩΤΗΣ ΠΑΝΑΓΙΩΤΟΥ, ΑΜ:1067517, email: up1067517@upnet.gr, 3ο Έτος
- ΔΑΝΑΗ ΧΑΛΟΥΛΟΥ, ΑΜ:1072596, email: up1072596@upnet.gr, 3ο Έτος
- ΚΩΝΣΤΑΝΤΙΝΟΣ ΠΑΡΑΣΚΕΥΟΠΟΥΛΟΣ, ΑΜ:1072608, email: up1072608@upnet.gr, 3ο Έτος
- ΣΟΦΙΑ ΛΑΜΠΡΟΠΟΥΛΟΥ, ΑΜ:1072606, email: up1072606@upnet.gr, 3ο Έτος


## Α. ΕΙΣΑΓΩΓΗ

Σε αυτή την εργαστηριακή άσκηση μας ζητήθηκε να υλοποιήσουμε ένα συντακτικό και έναν λεκτικό αναλυτή για το πρότυπο JSON. Το JSON αποτελεί πρότυπο το οποίο χρησιμοποιείται για την μετάδοση πληροφοριακών αντικειμένων δεδομένων. Στην συγκεκριμένη περίπτωση εξετάζουμε το JSON που χρησιμοποιεί ο ΟΠΑΠ για το ΤΖΟΚΕΡ. Για τον λεξικό αναλυτή χρησιμοποιήσαμε το πρόγραμμα flex ενώ για τον συντακτικό το πρόγραμμα bison. Για το BNF χρησιμοποιήσαμε το συντακτικό – γραμματική που αναγνωρίζει το εργαλείο bison.

## Β. BNF ΣΥΝΤΑΚΤΙΚΟΥ ΓΡΑΜΜΑΤΙΚΗΣ

*Το παρακάτω bnf είναι το συνολικό, και περιλαμβάνει και το bnf του πρώτου ερωτήματος.*

//root GRAMMAR

jsonfile: OBRACKETS ruleFULLJSON CBRACKETS;

ruleFULLJSON: ruleRANGE_RESULT

|ruleLAST_RESULT;

ruleLAST_RESULT: LAST COLON OBRACKETS ruleLAST_1 CBRACKETS COMMA ACTIVE COLON OBRACKETS ruleACTIVE CBRACKETS;

ruleLAST_1: ruleLAST_1 COMMA ruleLAST_1

|ruleGAME_ID

|ruleDRAW_ID

|ruleDRAW_TIME

|ruleSTATUS

|ruleDRAW_BREAK

|ruleVISUAL_DRAW

|rulePRICE_POINTS

|ruleWINNING_NUMS

|rulePRIZE_CATEG

|ruleWAGER_STATS

;

ruleACTIVE: ruleACTIVE COMMA ruleACTIVE

|ruleGAME_ID

|ruleDRAW_ID

|ruleDRAW_TIME

|ruleSTATUS

|ruleDRAW_BREAK

|ruleVISUAL_DRAW

|rulePRICE_POINTS

|rulePRIZE_CATEG

|ruleWAGER_STATS

;

ruleRANGE_RESULT: CONTENT COLON OBRACES ruleCONTENT_0 CBRACES COMMA ruleTOTALPAGES COMMA ruleTOTALELEMENTS COMMA ruleLAST COMMA ruleNUMBEROFELEMENTS COMMA SORT COLON OBRACES OBRACKETS ruleSORT CBRACKETS CBRACES COMMA ruleFIRST COMMA ruleSIZE COMMA ruleNUMBER;

ruleCONTENT_0: ruleCONTENT_1 COMMA ruleCONTENT_1 COMMA ruleCONTENT_1 COMMA ruleCONTENT_1 COMMA

;

```
ruleCONTENT_1: OBRACKETS ruleCONTENT CBRACKETS;

ruleCONTENT: ruleCONTENT COMMA ruleCONTENT
|ruleGAME_ID
|ruleDRAW_ID
|ruleDRAW_TIME
|ruleSTATUS
|ruleDRAW_BREAK
|ruleVISUAL_DRAW
|rulePRICE_POINTS
|ruleWINNING_NUMS
|rulePRIZE_CATEG
|ruleWAGER_STATS
;

ruleGAME_ID: GAME_ID COLON INT;

ruleDRAW_ID: DRAW_ID COLON INT;

ruleDRAW_TIME: DRAW_TIME COLON INT;

ruleSTATUS: STATUS COLON STRING;

ruleDRAW_BREAK: DRAW_BREAK COLON INT;

ruleVISUAL_DRAW: VISUAL_DRAW COLON INT;

rulePRICE_POINTS: PRICE_POINTS COLON OBRACKETS ruleAMOUNT CBRACKETS;

ruleAMOUNT: AMOUNT COLON DOUBLE;
/*-------------------------------------------------------------*/
```

ruleWINNING_NUMS: WINNING_NUMBERS COLON OBRACKETS ruleLIST COMMA ruleBONUS CBRACKETS;

ruleLIST: LIST COLON OBRACES DOUBLE COMMA DOUBLE COMMA INT CBRACES;

ruleBONUS: BONUS COLON OBRACES INT CBRACES;

/*----------*/

rulePRIZE_CATEG: PRIZE_CATEGORIES COLON rulePRIZE_ARRAY;

rulePRIZE_ARRAY: OBRACES ruleFIRST_PRIZE COMMA ruleREST_PRIZES COMMA ruleREST_PRIZES COMMA ruleREST_PRIZES COMMA ruleREST_PRIZES COMMA ruleREST_PRIZES COMMA ruleREST_PRIZES CBRACES;

ruleFIRST_PRIZE: OBRACKETS ruleID_1 COMMA ruleDIVIDENT COMMA ruleWINNERS COMMA ruleDISTRIBUTED COMMA ruleJACKPOT COMMA ruleFIXED COMMA ruleCATEG_TYPE COMMA ruleGAMETYPE COMMA ruleMIN_DIS CBRACKETS;

ruleREST_PRIZES: OBRACKETS ruleID COMMA ruleDIVIDENT COMMA ruleWINNERS COMMA ruleDISTRIBUTED COMMA ruleJACKPOT COMMA ruleFIXED COMMA ruleCATEG_TYPE COMMA ruleGAMETYPE CBRACKETS;

ruleID: ID COLON INT;

ruleID_1: ID COLON INT;

ruleDIVIDENT: DIVIDENT COLON DOUBLE

ruleWINNERS: WINNERS COLON INT

ruleDISTRIBUTED: DISTRIBUTED COLON DOUBLE

ruleJACKPOT: JACKPOT COLON DOUBLE

ruleFIXED: FIXED COLON DOUBLE

ruleCATEG_TYPE: CATEG_TYPE COLON INT

ruleGAMETYPE: GAMETYPE COLON STRING

ruleMIN_DIS: MINIMUM_DISTRIBUTED COLON DOUBLE

/*----------*/

ruleWAGER_STATS: WAGER_STATISTICS COLON OBRACKETS ruleCOLUMNS COMMA ruleWAGERS COMMA ruleADDON CBRACKETS;

```
ruleCOLUMNS: COLUMNS COLON INT;

ruleWAGERS: WAGERS COLON INT;

ruleADDON: ADDON COLON ruleJSONARRAY

ruleJSONARRAY: ruleEMPTYARRAY
|ruleARRAY;

ruleEMPTYARRAY: OBRACES CBRACES

ruleARRAY: OBRACES ruleCONTEXT CBRACES;

ruleCONTEXT: ruleCONTEXT COMMA ruleCONTEXT
|INT
|STRING
|DOUBLE
;

/*-------------------------------------------------------------------------------*/

ruleTOTALPAGES: TOTALPAGES COLON INT;
ruleTOTALELEMENTS: TOTALELEMENTS COLON INT;
ruleLAST: LAST COLON BOOLEAN;
ruleNUMBEROFELEMENTS: NUMBEROFELEMENTS COLON INT;

ruleSORT: ruleSORT COMMA ruleSORT
|ruleDIRECTION
|rulePROPERTY
|ruleIGNORECASE
```

|ruleNULLHANDLING

|ruleDESCENDING

|ruleASCENDING

;

ruleDIRECTION: DIRECTION COLON STRING;

rulePROPERTY: PROPERTY COLON STRING;

ruleIGNORECASE: IGNORECASE COLON BOOLEAN;

ruleNULLHANDLING: NULLHANDLING COLON STRING;

ruleDESCENDING: DESCENDING COLON BOOLEAN;

ruleASCENDING: ASCENDING COLON BOOLEAN;

/*-----------------------------*/

ruleFIRST: FIRST COLON BOOLEAN;

ruleSIZE: SIZE COLON INT;

ruleNUMBER: NUMBER COLON INT;

## Γ. ΚΩΔΙΚΑΣ FLEX

Παρακάτω φαίνεται ο κώδικας για το flex (είναι ο συνολικός κώδικας μαζί με το δεύτερο ερώτημα)

```
%{
    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>
    #include "2nd_Exc_Bison.tab.h"

    int line_num=1;


    void token_print(int token_id);

%}


%option yylineno
%option case-insensitive

int             [1-9][0-9]*|0
boolean         (true|false)
string          (\\.|[^\"])*
double      ([1-9][0-9]*|0).([0-9][0-9]*|0)
ws              [ \t\n\r]


%%
```

```
:                                   {return COLON;}
,                                   {return COMMA;}
\{                                  {return OBRACKETS;}
\}                                  {return CBRACKETS;}
\[                                  {return OBRACES;}
\]                                  {return CBRACES;}
\"last\"                    {return LAST;}
\"active\"                  {return ACTIVE;}
\"gameId\"                  {return GAME_ID;}
\"drawId\"                  {return DRAW_ID;}
\"drawTime\"                {return DRAW_TIME;}
\"status\"                  {return STATUS;}
\"drawBreak\"               {return DRAW_BREAK;}
\"visualDraw\"              {return VISUAL_DRAW;}
\"pricePoints\"             {return PRICE_POINTS;}
\"amount\"                  {return AMOUNT;}
\"winningNumbers\"          {return WINNING_NUMBERS;}
\"list\"                    {return LIST;}
\"bonus\"                   {return BONUS;}
\"prizeCategories\"         {return PRIZE_CATEGORIES;}
\"id\"                      {return ID;}
\"divident\"                {return DIVIDENT;}
\"winners\"                 {return WINNERS;}
\"distributed\"             {return DISTRIBUTED;}
\"jackpot\"                 {return JACKPOT;}
\"fixed\"                   {return FIXED;}
\"categoryType\"            {return CATEG_TYPE;}
\"gameType\"                {return GAMETYPE;}
\"minimumDistributed\"      {return MINIMUM_DISTRIBUTED;}
\"wagerStatistics\"         {return WAGER_STATISTICS;}
\"columns\"                 {return COLUMNS;}
\"wagers\"                  {return WAGERS;}
\"addOn\"                   {return ADDON;}

\"content\"                 {return CONTENT;}
\"totalPages\"              {return TOTALPAGES;}
\"totalElements\"           {return TOTALELEMENTS;}
\"numberOfElements\"        {return NUMBEROFELEMENTS;}
\"sort\"                    {return SORT;}
\"direction\"               {return DIRECTION;}
\"property\"                {return PROPERTY;}
\"ignoreCase\"              {return IGNORECASE;}
\"nullHandling\"            {return NULLHANDLING;}
\"descending\"              {return DESCENDING;}
\"ascending\"               {return ASCENDING;}
\"first\"                   {return FIRST;}
\"size\"                    {return SIZE;}
\"number\"                  {return NUMBER;}


{boolean}                   {return BOOLEAN;}
\"{string}\"                {return STRING;}
{int}                       {return INT;}
{double}                {return DOUBLE;}
\n                                  {++line_num;}
{ws}                        {;}


%%

int yywrap(void)
```

```
{
	return 1;
}
```

## Δ.ΚΩΔΙΚΑΣ BISON

Παρακάτω φαίνεται ο κώδικας για το bison (είναι ο συνολικός κώδικας μαζί με το δεύτερο ερώτημα)

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

extern FILE *yyin;
extern int yylex();
extern char* yytext;
extern int yyparse();
extern int line_num;
int errorline;
int errors=0;
int PrizeCatCounter=0;

int winningNums[5]; //Array where winning Numbers are going to be added
int win_array_index=0;

//Error Functions
void incPrizeCatCntr(void); //Add +1 to PrizeCatCounter after each "id" in
"PrizeCategories" and break operation if >8
void addNumber(char* n); //Add current winning Number to winningNums[5]
array
void checkGameID(char* l); //Function to Check GameID
void checkWinningNumbers(int winArray[5]); //Function to Check if Winning
Numbers are exactly 5 and between 1 and 45
void checkCatType(char* p); //Check if Category Type is 0 or 1

void yyerror(const char* s);
%}

%define parse.error verbose
%union {
int num;
char* string;}
%token LAST ACTIVE
%token COLON COMMA OBRACKETS OBRACES CBRACKETS CBRACES
%token GAME_ID GAME_ID_ATTRIBUTE
%token DRAW_ID DRAW_TIME
%token STATUS
%token DRAW_BREAK VISUAL_DRAW
%token PRICE_POINTS     AMOUNT
%token WINNING_NUMBERS LIST
%token BONUS PRIZE_CATEGORIES
%token ID DIVIDENT
%token WINNERS DISTRIBUTED
%token JACKPOT FIXED CATEG_TYPE GAMETYPE MINIMUM_DISTRIBUTED
WAGER_STATISTICS COLUMNS
%token WAGERS
%token ADDON
%token BOOLEAN
%token STRING
```

```
%token INT
%token DOUBLE

%token CONTENT
%token TOTALPAGES
%token TOTALELEMENTS
%token NUMBEROFELEMENTS
%token SORT
%token DIRECTION
%token PROPERTY
%token IGNORECASE
%token NULLHANDLING
%token DESCENDING
%token ASCENDING
%token FIRST
%token SIZE
%token NUMBER



%type <double> DOUBLE
%type <string> STRING
%type <num> INT
%type <boolean> BOOLEAN
%start jsonfile

%%
//root GRAMMAR
jsonfile: OBRACKETS {printf("{\n");} ruleFULLJSON CBRACKETS
{printf("}\n");};

ruleFULLJSON: ruleRANGE_RESULT
|ruleLAST_RESULT;


ruleLAST_RESULT: LAST {printf("\"last\"");} COLON {printf(":");} OBRACKETS
{printf("{");} ruleLAST_1 CBRACKETS {printf("}");} COMMA {printf(",\n");}
ACTIVE {printf("\"active\"");} COLON {printf(":");} OBRACKETS
{printf("{");} ruleACTIVE CBRACKETS {printf("}");};

ruleRANGE_RESULT: CONTENT {printf("\"content\"");} COLON {printf(":");}
OBRACES {printf("[");}ruleCONTENT_0 CBRACES {printf("]");} COMMA
{printf(",\n");} ruleTOTALPAGES COMMA {printf(",\n");} ruleTOTALELEMENTS
COMMA {printf(",\n");} ruleLAST COMMA {printf(",\n");} ruleNUMBEROFELEMENTS
COMMA {printf(",\n");} SORT  {printf("\"sort\"");} COLON {printf(":");}
OBRACES {printf("[");} OBRACKETS {printf("{");} ruleSORT CBRACKETS
{printf("}");} CBRACES {printf("]");} COMMA {printf(",\n");} ruleFIRST
COMMA {printf(",\n");} ruleSIZE COMMA {printf(",\n");} ruleNUMBER;

ruleCONTENT_0: ruleCONTENT_1 COMMA {printf(",\n");} ruleCONTENT_1 COMMA
{printf(",\n");} ruleCONTENT_1 COMMA {printf(",\n");} ruleCONTENT_1 COMMA
{printf(",\n");};

ruleCONTENT_1: OBRACKETS ruleCONTENT CBRACKETS;

ruleCONTENT: ruleCONTENT COMMA {printf(",\n");} ruleCONTENT
|ruleGAME_ID
|ruleDRAW_ID
|ruleDRAW_TIME
|ruleSTATUS
|ruleDRAW_BREAK
```

```
|ruleVISUAL_DRAW
|rulePRICE_POINTS
|ruleWINNING_NUMS
|rulePRIZE_CATEG
|ruleWAGER_STATS
;

ruleACTIVE: ruleACTIVE COMMA {printf(",\n");} ruleACTIVE
|ruleGAME_ID
|ruleDRAW_ID
|ruleDRAW_TIME
|ruleSTATUS
|ruleDRAW_BREAK
|ruleVISUAL_DRAW
|rulePRICE_POINTS
|rulePRIZE_CATEG
|ruleWAGER_STATS
;

ruleLAST_1: ruleLAST_1 COMMA {printf(",\n");} ruleLAST_1
|ruleGAME_ID
|ruleDRAW_ID
|ruleDRAW_TIME
|ruleSTATUS
|ruleDRAW_BREAK
|ruleVISUAL_DRAW
|rulePRICE_POINTS
|ruleWINNING_NUMS
|rulePRIZE_CATEG
|ruleWAGER_STATS
;

ruleGAME_ID: GAME_ID {printf("\"gameId\"");} COLON {printf(":");} INT
{checkGameID(yytext);printf("\%s", yytext);};

ruleDRAW_ID: DRAW_ID {printf("\"drawId\"");} COLON {printf(":");} INT
{printf("\%s", yytext);} ;

ruleDRAW_TIME: DRAW_TIME {printf("\"drawTime\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};


ruleSTATUS: STATUS {printf("\"status\"");} COLON {printf(":");}
ruleSTATUS_2;

ruleSTATUS_2: STRING {printf("\%s", yytext);}
|ACTIVE {printf("\%s", yytext);};


ruleDRAW_BREAK: DRAW_BREAK {printf("\"drawBreak\"");} COLON {printf(":");}
INT {printf("\%s", yytext);};

ruleVISUAL_DRAW: VISUAL_DRAW {printf("\"visualDraw\"");} COLON
{printf(":");} INT {printf("\%s", yytext);};

rulePRICE_POINTS: PRICE_POINTS {printf("\"pricePoints\"");} COLON
{printf(":");} OBRACKETS {printf("{\n");} ruleAMOUNT CBRACKETS
{printf("}\n");};

ruleAMOUNT: AMOUNT {printf("\"amount\"");} COLON {printf(":");} DOUBLE
{printf("\%s", yytext);};
```

```
/*-------------------------------------------------------------*/
ruleWINNING_NUMS: WINNING_NUMBERS {printf("\"winningNumbers\"");} COLON
{printf(":");} OBRACKETS {printf("{\n");} ruleLIST COMMA {printf(",\n");}
ruleBONUS CBRACKETS {printf("}\n");};

ruleLIST: LIST {printf("\"list\"");} COLON {printf(":");} OBRACES
{printf("[\n");} INT {addNumber(yytext);printf("\%s", yytext);} COMMA
{printf(",\n");} INT {addNumber(yytext);printf("\%s", yytext);} COMMA
{printf(",\n");} INT {addNumber(yytext);printf("\%s", yytext);} COMMA
{printf(",\n");} INT {addNumber(yytext);printf("\%s", yytext);} COMMA
{printf(",\n");} INT {addNumber(yytext);printf("\%s",
yytext);checkWinningNumbers(winningNums); win_array_index=0;} CBRACES
{printf("]\n");};

ruleBONUS: BONUS {printf("\"bonus\"");} COLON {printf(":");} OBRACES
{printf("[\n");} INT {printf("\%s", yytext);} CBRACES {printf("]\n");};
/*-------------------------------------------------------------*/
rulePRIZE_CATEG: PRIZE_CATEGORIES {printf("\"prizeCategories\"");} COLON
{printf(":");} rulePRIZE_ARRAY;

rulePRIZE_ARRAY: OBRACES {printf("[\n");} ruleFIRST_PRIZE COMMA
{printf(",\n");} ruleREST_PRIZES COMMA {printf(",\n");} ruleREST_PRIZES
COMMA {printf(",\n");} ruleREST_PRIZES COMMA {printf(",\n");}
ruleREST_PRIZES COMMA {printf(",\n");} ruleREST_PRIZES COMMA
{printf(",\n");} ruleREST_PRIZES COMMA {printf(",\n");} ruleREST_PRIZES
CBRACES {PrizeCatCounter=0; printf("]\n");};

ruleFIRST_PRIZE: OBRACKETS {printf("{\n");} ruleID_1 COMMA {printf(",\n");}
ruleDIVIDENT COMMA {printf(",\n");} ruleWINNERS COMMA {printf(",\n");}
ruleDISTRIBUTED COMMA {printf(",\n");} ruleJACKPOT COMMA {printf(",\n");}
ruleFIXED COMMA {printf(",\n");} ruleCATEG_TYPE COMMA {printf(",\n");}
ruleGAMETYPE COMMA {printf(",\n");} ruleMIN_DIS CBRACKETS {printf("}\n");};

ruleREST_PRIZES: OBRACKETS {printf("{\n");} ruleID COMMA {printf(",\n");}
ruleDIVIDENT COMMA {printf(",\n");} ruleWINNERS COMMA {printf(",\n");}
ruleDISTRIBUTED COMMA {printf(",\n");} ruleJACKPOT COMMA {printf(",\n");}
ruleFIXED COMMA {printf(",\n");} ruleCATEG_TYPE COMMA {printf(",\n");}
ruleGAMETYPE CBRACKETS {printf("}\n");};

ruleID: ID {incPrizeCatCntr();printf("\"id\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};
ruleID_1: ID {incPrizeCatCntr();printf("\"id\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};
ruleDIVIDENT: DIVIDENT {printf("\"divident\"");} COLON {printf(":");}
DOUBLE {printf("\%s", yytext);}
ruleWINNERS: WINNERS {printf("\"winners\"");} COLON {printf(":");} INT
{printf("\%s", yytext);}
ruleDISTRIBUTED: DISTRIBUTED {printf("\"distributed\"");} COLON
{printf(":");} DOUBLE {printf("\%s", yytext);}
ruleJACKPOT: JACKPOT {printf("\"jackpot\"");} COLON {printf(":");} DOUBLE
{printf("\%s", yytext);}
ruleFIXED: FIXED {printf("\"fixed\"");} COLON {printf(":");} DOUBLE
{printf("\%s", yytext);}
ruleCATEG_TYPE: CATEG_TYPE {printf("\"categoryType\"");} COLON
{printf(":");} INT {checkCatType(yytext);printf("\%s", yytext);}
ruleGAMETYPE: GAMETYPE {printf("\"gameType\"");} COLON {printf(":");}
STRING {printf("\%s", yytext);}
ruleMIN_DIS: MINIMUM_DISTRIBUTED {printf("\"minimumDistributed\"");} COLON
{printf(":");} DOUBLE {printf("\%s", yytext);}
/*-------------------------------------------------------------*/
ruleWAGER_STATS: WAGER_STATISTICS {printf("\"wagerStatistics\"");} COLON
```

```
{printf(":");} OBRACKETS {printf("{\n");} ruleCOLUMNS COMMA {printf(",");}
ruleWAGERS COMMA {printf(",");} ruleADDON CBRACKETS {printf("}\n");};

ruleCOLUMNS: COLUMNS {printf("\"columns\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};

ruleWAGERS: WAGERS {printf("\"wagers\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};

ruleADDON: ADDON {printf("\"addOn\"");} COLON {printf(":");} ruleJSONARRAY

ruleJSONARRAY: ruleEMPTYARRAY
|ruleARRAY;

ruleEMPTYARRAY: OBRACES {printf("[\n");} CBRACES {printf("]\n");};

ruleARRAY: OBRACES {printf("[\n");} ruleCONTEXT CBRACES {printf("]\n");};

ruleCONTEXT: ruleCONTEXT COMMA {printf(",\n");} ruleCONTEXT
|INT {printf("\%s", yytext);}
|STRING {printf("\%s", yytext);}
|DOUBLE {printf("\%s", yytext);}
;

/*-------------------------------------------------------------------------
-------*/

ruleTOTALPAGES: TOTALPAGES {printf("\"totalPages\"");} COLON {printf(":");}
INT {printf("\%s", yytext);};
ruleTOTALELEMENTS: TOTALELEMENTS {printf("\"totalElements\"");} COLON
{printf(":");} INT {printf("\%s", yytext);};
ruleLAST: LAST {printf("\"last\"");} COLON {printf(":");} BOOLEAN
{printf("\%s", yytext);};
ruleNUMBEROFELEMENTS: NUMBEROFELEMENTS {printf("\"numberOfElements\"");}
COLON {printf(":");} INT {printf("\%s", yytext);};


ruleSORT: ruleSORT COMMA {printf(",\n");} ruleSORT
|ruleDIRECTION
|rulePROPERTY
|ruleIGNORECASE
|ruleNULLHANDLING
|ruleDESCENDING
|ruleASCENDING
;
ruleDIRECTION: DIRECTION {printf("\"direction\"");} COLON {printf(":");}
STRING {printf("\%s", yytext);};
rulePROPERTY: PROPERTY {printf("\"property\"");} COLON {printf(":");}
STRING {printf("\%s", yytext);};
ruleIGNORECASE: IGNORECASE {printf("\"ignoreCase\"");} COLON {printf(":");}
BOOLEAN {printf("\%s", yytext);};
ruleNULLHANDLING: NULLHANDLING {printf("\"nullHandling\"");} COLON
{printf(":");} STRING {printf("\%s", yytext);};
ruleDESCENDING: DESCENDING {printf("\"descending\"");} COLON {printf(":");}
BOOLEAN {printf("\%s", yytext);};
ruleASCENDING: ASCENDING {printf("\"ascending\"");} COLON {printf(":");}
BOOLEAN {printf("\%s", yytext);};
/*------------------------------*/
ruleFIRST: FIRST {printf("\"first\"");} COLON {printf(":");} BOOLEAN
{printf("\%s", yytext);};
ruleSIZE: SIZE {printf("\"size\"");} COLON {printf(":");} INT
```

```
{printf("\%s", yytext);};
ruleNUMBER: NUMBER {printf("\"number\"");} COLON {printf(":");} INT
{printf("\%s", yytext);};



%%

void incPrizeCatCntr(void)
{
	PrizeCatCounter++;
	if(PrizeCatCounter>8){
		errors++;
		printf("\nError in line %i \nMore than 8 PrizeCategories",
line_num);
		exit(EXIT_FAILURE);
	}
}

void addNumber(char* n)
{
	if(win_array_index>4){
		errors++;
		printf("\nError in line %i \nMore than 5 Winning Numbers",
line_num);
		exit(EXIT_FAILURE);
	}
	else{
	winningNums[win_array_index]=atoi(n);
	win_array_index++;
	}
}

void checkGameID(char* l)
{
		int gameid;
		gameid=atoi(l);
		printf("GameId=%d\n",gameid);

	if(gameid!=1100&&gameid!=1110&&gameid!=2100&&gameid!=2101&&gameid!=51
03&&gameid!=5104&&gameid!=5106){
			errors++;
			printf("\nError in line %i \nWrong GameID", line_num);
			exit(EXIT_FAILURE);
		}

}



void checkWinningNumbers(int Array[]){
	int i;
	for(i=0; i<5; i++){
		if(Array[i]<1||Array[i]>45){
		errors++;
		printf("\nError in line %i \nWinning Numbers Bigger than 45 or
less than 1", line_num);
		exit(EXIT_FAILURE);
		}
	}
}
```

```
void checkCatType(char* p){
	if(atoi(p)!=0&&atoi(p)!=1){
		errors++;
		printf("\nError in line %i \nIncorrect Category Type",
line_num);
		exit(EXIT_FAILURE);
	}
}

int main (int argc, char **argv) {
	FILE *jfile = fopen(argv[1], "r");
	yyin = jfile;
	yyparse();
	if(errors!=0)
	{
		printf("\nError in line %i", errorline);
		exit(EXIT_FAILURE);}
	else
	{
		printf("\n\n\t\tSuccesful Parse!\n");
		return 0;
	}
	}

void yyerror (const char *s) {fprintf (stderr, "%s\nin line: %d\nlast
scanned: %s\n", s, line_num, yytext);}
```

## Ε. Εξήγηση Συναρτήσεων

Χρησιμοποιήσαμε τη συνάρτηση incPrizeCatCntr(void) η οποία αυξάνει κατά 1 τη μεταβλητή PrizeCatCounter μετά την αναγνώριση του id στο PrizeCategories ενώ σταματάει το parsing εάν αυτή η μεταβλητή είναι μεγαλύτερη του 8.

```
void incPrizeCatCntr(void)
{
	PrizeCatCounter++;
	if(PrizeCatCounter>8){
		errors++;
		printf("\nError in line %i \nMore than 8 PrizeCategories",
line_num);
		exit(EXIT_FAILURE);
	}
}
```

Χρησιμοποιήσαμε, επίσης, τη συνάρτηση `addNumber(char* n)` η οποία παίρνει σαν όρισμα αριθμούς από το πεδίο list (αριθμοί που κληρώθηκαν) και τους προσθέτει στον πίνακα winningNums.

```
void addNumber(char* n)
{
	if(win_array_index>4){
		errors++;
		printf("\nError in line %i \nMore than 5 Winning Numbers",
line_num);
		exit(EXIT_FAILURE);
```

```
        }
        else{
        winningNums[win_array_index]=atoi(n);
        win_array_index++;
        }
}
```

Χρησιμοποιήσαμε, επιπροσθέτως, τη συνάρτηση `checkWinningNumbers(int Array[])` η οποία ελέγχει αν οι αριθμοί που κληρώθηκαν είναι ακριβώς 5 και ανάμεσα στο 1 και στο 45.

```
void checkWinningNumbers(int Array[]){
        int i;
        for(i=0; i<5; i++){
                if(Array[i]<1||Array[i]>45){
                errors++;
                printf("\nError in line %i \nWinning Numbers Bigger than 45 or
less than 1", line_num);
                exit(EXIT_FAILURE);
                }
        }
}
```

Επιπλέον, χρησιμοποιήσαμε τη συνάρτηση `checkGameID(char* l)` η οποία παίρνει σαν όρισμα το gameid και ελέγχει αν είναι ένα από τα επιτρεπτά.

```
void checkGameID(char* l)
{
                int gameid;
                gameid=atoi(l);
                printf("GameId=%d\n",gameid);

        if(gameid!=1100&&gameid!=1110&&gameid!=2100&&gameid!=2101&&gameid!=51
03&&gameid!=5104&&gameid!=5106){
                        errors++;
                        printf("\nError in line %i \nWrong GameID", line_num);
                        exit(EXIT_FAILURE);
                }

}
```

Επιπρόσθετα, χρησιμοποιήσαμε τη συνάρτηση `checkCatType(char* p)` η οποία ελέγχει αν το categoryType είναι 0 ή 1.

```
void checkCatType(char* p){
        if(atoi(p)!=0&&atoi(p)!=1){
                errors++;
                printf("\nError in line %i \nIncorrect Category Type",
line_num);
                exit(EXIT_FAILURE);
        }
}
```

Τέλος, χρησιμοποιήσαμε τη συνάρτηση `yyerror (const char *s)` η οποία εμφανίζει το μήνυμα λάθους σε περίπτωση που αυτό βρεθεί καθώς και τη γραμμή που αυτό βρίσκεται.

```
void yyerror (const char *s) {fprintf (stderr, "%s\nin line: %d\nlast
```

```
scanned: %s\n", s, line_num, yytext);}
```

Το **%define parse.error verbose** είναι οδηγία του bison που εμφανίζει πιο συγκεκριμένες πληροφορίες σε περίπτωση εμφάνισης σφάλματος.

Η μεταβλητή **yytext** περιέχει το κομμάτι του αρχείου που γίνεται parse εκείνη τη στιγμή και τη χρησιμοποιούμε για να κάνουμε print ορισμένες μεταβλητές.

Στις εκτελέσεις παρακάτω εμφανίζονται ορισμένα warnings τύπου shift/reduce, reduce/reduce. Ωστόσο, αυτά δεν επηρεάζουν την ομαλή λειτουργία του συντακτικού αναλυτή.

## ΣΤ. ΑΡΧΕΙΑ ΕΙΣΟΔΟΥ

ΑΡΧΕΙΑ ΜΕ ΣΦΑΛΜΑΤΑ

Παρακάτω εμφανίζεται ένα λανθασμένο JSON αρχείο. Το λάθος του βρίσκεται στο ότι στη λίστα winningNumbers έχουμε αντικαταστήσει μια αποδεκτή τιμή με μια μη αποδεκτή εφόσον ξεπερνάει τον αριθμό 45.


```
{
    "last": {
        "gameId": 5104,
        "drawId": 2390,
        "drawTime": 1642363200000,
        "status": "results",
        "drawBreak": 1800000,
        "visualDraw": 2390,
        "pricePoints": {
            "amount": 0.5
        },
        "winningNumbers": {
            "list": [
                1,
                29,
                26,
                24,
                50
            ],
            "bonus": [
                6
            ]
        },
        "prizeCategories": [
            {
                "id": 1,
                "divident": 0.0,
                "winners": 0,
                "distributed": 356871.53,
                "jackpot": 748954.15,
                "fixed": 0.0,
```

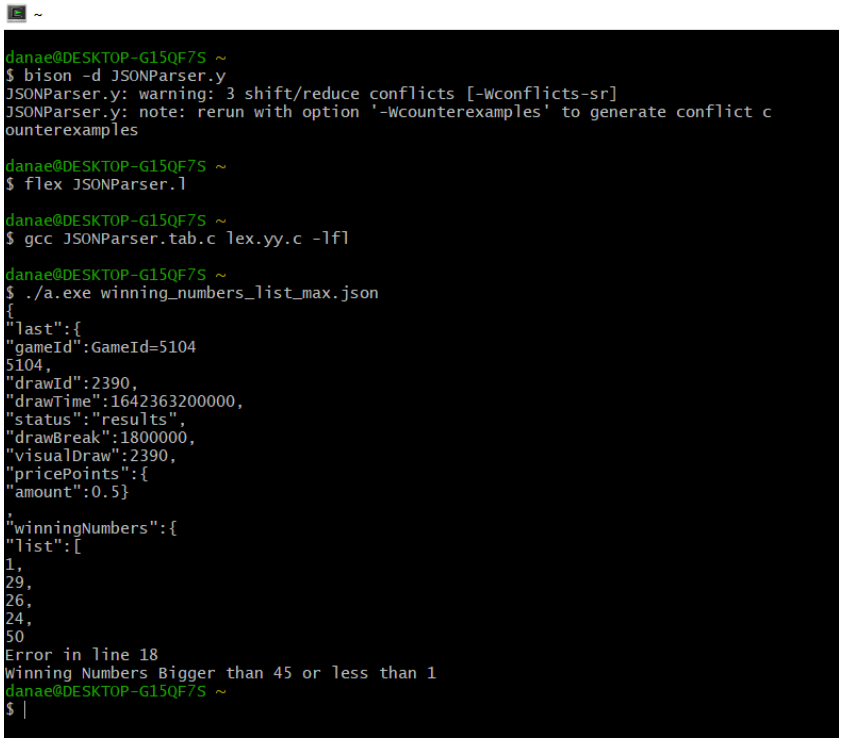
```
            "categoryType": 0,
            "gameType": "Normal",
            "minimumDistributed": 0.0
        },
        {
            "id": 2,
            "divident": 22575.97,
            "winners": 4,
            "distributed": 55178.93,
            "jackpot": 35124.97,
            "fixed": 0.0,
            "categoryType": 0,
            "gameType": "Normal"
        },
        {
            "id": 3,
            "divident": 2500.0,
            "winners": 20,
            "distributed": 50000.0,
            "jackpot": 0.0,
            "fixed": 2500.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 4,
            "divident": 50.0,
            "winners": 326,
            "distributed": 16300.0,
            "jackpot": 0.0,
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 5,
            "divident": 50.0,
            "winners": 816,
            "distributed": 40800.0,
            "jackpot": 0.0,
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 6,
            "divident": 2.0,
            "winners": 16634,
```

```
                "distributed": 33268.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 7,
                "divident": 2.0,
                "winners": 10341,
                "distributed": 20682.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 8,
                "divident": 1.5,
                "winners": 49233,
                "distributed": 73849.5,
                "jackpot": 0.0,
                "fixed": 1.5,
                "categoryType": 1,
                "gameType": "Normal"
            }
        ],
        "wagerStatistics": {
            "columns": 2866438,
            "wagers": 503579,
            "addOn": []
        }
    },
    "active": {
        "gameId": 5104,
        "drawId": 2391,
        "drawTime": 1642536000000,
        "status": "active",
        "drawBreak": 1800000,
        "visualDraw": 2391,
        "pricePoints": {
            "amount": 0.5
        },
        "prizeCategories": [
            {
                "id": 1,
                "divident": 0.0,
                "winners": 0,
```

```
            "distributed": 0.0,
            "jackpot": 1105825.68,
            "fixed": 0.0,
            "categoryType": 0,
            "gameType": "Normal",
            "minimumDistributed": 1300000.0
        },
        {
            "id": 2,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 0.0,
            "categoryType": 0,
            "gameType": "Normal"
        },
        {
            "id": 3,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 2500.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 4,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 5,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
```

```
                "id": 6,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 7,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 8,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 1.5,
                "categoryType": 1,
                "gameType": "Normal"
            }
        ],
        "wagerStatistics": {
            "columns": 0,
            "wagers": 0,
            "addOn": []
        }
    }
}
```


Για το αρχείο winning_numbers_list_max




Παρακάτω εμφανίζεται ένα λανθασμένο JSON αρχείο. Το λάθος του βρίσκεται στο ότι στο πεδίο distibuted έχουμε βάλει μια ακέραια τιμή ενώ θα έπρεπε να παίρνει μόνο τύπου double τιμές που είναι μη αποδεκτή.


```
{
    "last": {
        "gameId": 5104,
        "drawId": 2390,
        "drawTime": 1642363200000,
        "status": "results",
        "drawBreak": 1800000,
        "visualDraw": 2390,
        "pricePoints": {
            "amount": 0.5
        },
        "winningNumbers": {
            "list": [
                1,
                29,
                26,
                24,
                17
            ],
            "bonus": [
                6
            ]
        },
```

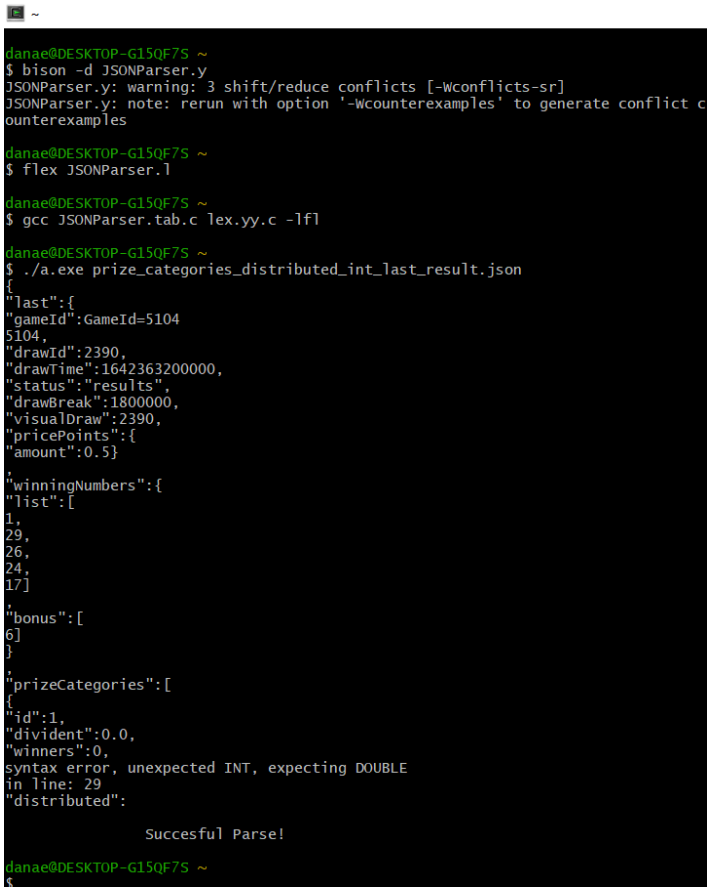
```
        "prizeCategories": [
            {
                "id": 1,
                "divident": 0.0,
                "winners": 0,
                "distributed": 356871,
                "jackpot": 748954.15,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal",
                "minimumDistributed": 0.0
            },
            {
                "id": 2,
                "divident": 22575.97,
                "winners": 4,
                "distributed": 55178.93,
                "jackpot": 35124.97,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal"
            },
            {
                "id": 3,
                "divident": 2500.0,
                "winners": 20,
                "distributed": 50000.0,
                "jackpot": 0.0,
                "fixed": 2500.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 4,
                "divident": 50.0,
                "winners": 326,
                "distributed": 16300.0,
                "jackpot": 0.0,
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 5,
                "divident": 50.0,
                "winners": 816,
                "distributed": 40800.0,
                "jackpot": 0.0,
```

```
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 6,
                "divident": 2.0,
                "winners": 16634,
                "distributed": 33268.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 7,
                "divident": 2.0,
                "winners": 10341,
                "distributed": 20682.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 8,
                "divident": 1.5,
                "winners": 49233,
                "distributed": 73849.5,
                "jackpot": 0.0,
                "fixed": 1.5,
                "categoryType": 1,
                "gameType": "Normal"
            }
        ],
        "wagerStatistics": {
            "columns": 2866438,
            "wagers": 503579,
            "addOn": []
        }
    },
    "active": {
        "gameId": 5104,
        "drawId": 2391,
        "drawTime": 1642536000000,
        "status": "active",
        "drawBreak": 1800000,
        "visualDraw": 2391,
```

```
        "pricePoints": {
            "amount": 0.5
        },
        "prizeCategories": [
            {
                "id": 1,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 1105825.68,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal",
                "minimumDistributed": 1300000.0
            },
            {
                "id": 2,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal"
            },
            {
                "id": 3,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 2500.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 4,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 5,
                "divident": 0.0,
```

```
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 6,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 7,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 8,
                "divident": 0.0,
                "winners": 0,
                "distributed": 0.0,
                "jackpot": 0.0,
                "fixed": 1.5,
                "categoryType": 1,
                "gameType": "Normal"
            }
        ],
        "wagerStatistics": {
            "columns": 0,
            "wagers": 0,
            "addOn": []
        }
    }
}
```

Για το αρχείο prize_categories_distributed_int_last_result




Παρακάτω εμφανίζεται ένα λανθασμένο JSON αρχείο. Το λάθος του βρίσκεται στο ότι στο πεδίο gameid έχουμε βάλει την τιμή 3600 που είναι μη αποδεκτή.


```
{
    "last": {
        "gameId": 3600,
        "drawId": 2390,
        "drawTime": 1642363200000,
        "status": "results",
        "drawBreak": 1800000,
        "visualDraw": 2390,
        "pricePoints": {
            "amount": 0.5
        },
        "winningNumbers": {
            "list": [
                1,
                29,
                26,
```

```
                24,
                17
            ],
            "bonus": [
                6
            ]
        },
        "prizeCategories": [
            {
                "id": 1,
                "divident": 0.0,
                "winners": 0,
                "distributed": 356871.53,
                "jackpot": 748954.15,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal",
                "minimumDistributed": 0.0
            },
            {
                "id": 2,
                "divident": 22575.97,
                "winners": 4,
                "distributed": 55178.93,
                "jackpot": 35124.97,
                "fixed": 0.0,
                "categoryType": 0,
                "gameType": "Normal"
            },
            {
                "id": 3,
                "divident": 2500.0,
                "winners": 20,
                "distributed": 50000.0,
                "jackpot": 0.0,
                "fixed": 2500.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 4,
                "divident": 50.0,
                "winners": 326,
                "distributed": 16300.0,
                "jackpot": 0.0,
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
```

```
            },
            {
                "id": 5,
                "divident": 50.0,
                "winners": 816,
                "distributed": 40800.0,
                "jackpot": 0.0,
                "fixed": 50.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 6,
                "divident": 2.0,
                "winners": 16634,
                "distributed": 33268.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 7,
                "divident": 2.0,
                "winners": 10341,
                "distributed": 20682.0,
                "jackpot": 0.0,
                "fixed": 2.0,
                "categoryType": 1,
                "gameType": "Normal"
            },
            {
                "id": 8,
                "divident": 1.5,
                "winners": 49233,
                "distributed": 73849.5,
                "jackpot": 0.0,
                "fixed": 1.5,
                "categoryType": 1,
                "gameType": "Normal"
            }
        ],
        "wagerStatistics": {
            "columns": 2866438,
            "wagers": 503579,
            "addOn": []
        }
    },
```

```
"active": {
    "gameId": 5104,
    "drawId": 2391,
    "drawTime": 1642536000000,
    "status": "active",
    "drawBreak": 1800000,
    "visualDraw": 2391,
    "pricePoints": {
        "amount": 0.5
    },
    "prizeCategories": [
        {
            "id": 1,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 1105825.68,
            "fixed": 0.0,
            "categoryType": 0,
            "gameType": "Normal",
            "minimumDistributed": 1300000.0
        },
        {
            "id": 2,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 0.0,
            "categoryType": 0,
            "gameType": "Normal"
        },
        {
            "id": 3,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 2500.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 4,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
```

```
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 5,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 50.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 6,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 2.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 7,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 2.0,
            "categoryType": 1,
            "gameType": "Normal"
        },
        {
            "id": 8,
            "divident": 0.0,
            "winners": 0,
            "distributed": 0.0,
            "jackpot": 0.0,
            "fixed": 1.5,
            "categoryType": 1,
            "gameType": "Normal"
        }
    ],
    "wagerStatistics": {
        "columns": 0,
        "wagers": 0,
```

```
            "addOn": []
        }
    }
}
```

Για το αρχείο game_id_wrong_number

```
danae@DESKTOP-G15QF7S ~
$ ./a.exe game_id_wrong_number.json
{
"last":{
"gameId":GameId=3600

Error in line 3
Wrong GameID
danae@DESKTOP-G15QF7S ~
$
```

## Z. ΠΑΡΑΔΕΙΓΜΑΤΑ ΣΩΣΤΗΣ ΕΚΤΕΛΕΣΗΣ

### Range Result & Last Result

```
Administrator@MyPC ~/2022/Prog_Lang_Proj/FINAL_CODES/PROJECT/2nd_Exc
$ ./a.exe last_result.json
{
"last":{"gameId":GameId=5104
5104,
"drawId":2390,
"drawTime":1642363200000,
"status":"results",
"drawBreak":1800000,
"visualDraw":2390,
"pricePoints":{
"amount":0.5}
,
"winningNumbers":{
"list":[
1,
29,
26,
24,
17]
,
"bonus":[
6]
}
,
"prizeCategories":[
{
"id":1,
"divident":0.0,
"winners":0,
"distributed":356871.53,
"jackpot":748954.15,
"fixed":0.0,
"categoryType":0,
"gameType":"Normal",
"minimumDistributed":0.0}
,
{
"id":2,
"divident":22575.97,
"winners":4,
```

```
,
{
"id":7,
"divident":0.0,
"winners":0,
"distributed":0.0,
"jackpot":0.0,
"fixed":2.0,
"categoryType":1,
"gameType":"Normal"}
,
{
"id":8,
"divident":0.0,
"winners":0,
"distributed":0.0,
"jackpot":0.0,
"fixed":1.5,
"categoryType":1,
"gameType":"Normal"}
]
,
"wagerStatistics":{
"columns":0,"wagers":0,"addOn":[
]
}
}}


                Succesful Parse!
```


Ο κώδικας θα γίνει Compile με τις εντολές:
Για την Άσκηση 1 + 10%:
***bison -d JSONParser.y***
***flex JSONParser.l***
***gcc JSONParser.tab.c lex.yy.c -lfl***
***./a.exe last_result.json***
Για την Άσκηση 2:
***bison -d 2nd_Exc_Bison.y***
***flex 2nd_Exc_Flex.l***
***gcc 2nd_Exc_Bison.tab.c lex.yy.c -lfl***
***./a.exe range_result.json***
***./a.exe last_result.json***