

Project Flex/Bison

Αρχές Γλωσσών Προγραμματισμού και

Μεταφραστών

Όνομα/ΑΜ: Περικλής Κοροντζής / 1072563
Γρηγόριος Γεώργιος Καπαδούκας / 1072484
Ηλίας Σερταρίδης / 1072480

Έτος: 2021 – 2022

Εξάμηνο: 6^ο

Email: up1072563@upnet.gr
up1072484@upnet.gr
up1072480@upnet.gr

Online link: <https://github.com/PeriKor19008/compiler>

BNF Ερωτήματος 1:

```
<compiler> ::= { <last> , <active> }  
<last> ::= { "last":{ <gameId> , <drawId> ,  
                    <drawTime> , <drawTime> , <status> ,  
                    <drawBrea> , <visualDraw> ,  
                    <pricePoints> , <winningNumbers> ,  
                    <prizeCategories> , <wagerStatistics> }  
<active> ::= { "active": { <gameId> , <drawId> ,  
                          <drawTime> , <status> , <drawBreak> ,
```

	<visualDraw> , <pricePoints> , <prizeCategories> , <wagerStatistics> }
<gameId>	::= "gameId": <θετικός ακέραιος>
<drawId>	::= "drawId": <θετικός ακέραιος>
<drawTime>	::= "drawTime": <θετικός ακέραιος>
<status>	::= "status": <αλφαριθμητικό>
<drawBreak>	::= "drawBreak": <θετικός ακέραιος>
<visualDraw>	::= "visualDraw": <θετικός ακέραιος>
<pricePoints>	::= "pricePoints": { <amount> }
<amount>	::= "amount": <θετικός πραγματικός>
<winningNumber>	::= "winningNumbers": { <list> , <bonus> }
<list>	::= "list": <array θετικού ακέραιου>
<bonus>	::= "bonus": <array θετικών ακεραίων>
<prizeCategories>	::= "prizeCategories": ["id" = 1 , <dividend> <winners> , <distributed> , <jackpot> , <fixed> , <categoryType> , <gameType> , <minimumDistributed>] "prizeCategories": [<id> , <dividend> , <winners> , <distributed> , <jackpot> , <fixed> , <categoryType> , <gameType> , <minimumDistributed>]
<id>	::= "id": <θετικός ακέραιος εύρους 1-8>
<dividend>	::= "dividend": <θετικός πραγματικός>
<winners>	::= "winners": <θετικός ακέραιος>
<distributed>	::= "distributed": <boolean>
<jackpot>	::= "distributed": <θετικός πραγματικός>
<fixed>	::= "fixed": <θετικός πραγματικός>
<categoryType>	::= "categoryType": <θετικός ακέραιος 0 ή 1>
<gameType>	::= "gameType": <αλφαριθμητικό>

<minimumDistributed>	::=	"minimumDistributed": <θετικός πραγματικός>
<wagerStatistics>	::=	"wagerStatistics": { <columns> , <wagers> , <addOn> }
<columns>	::=	"columns": <θετικός ακέραιος>
<wagers>	::=	"wagers": <θετικός ακέραιος>
<addOn>	::=	"addOn": <JSON array>
<θετικός ακέραιος>	::=	(0 1 2 3 4 5 6 7 8 9) ⁺
<αλφαριθμητικό>	::=	(a b c ... y z) ⁺
<θετικός πραγματικός>	::=	(<θετικός ακέραιος>) . (<θετικός ακέραιος>)
<array θετικών ακεραίων>	::=	[(<θετικός ακέραιος> ,) ⁺ <θετικός ακέραιος>
<array θετικού ακεραίου>	::=	[<θετικός ακέραιος>]
<θετικός ακέραιος εύρους 2-8>	::=	2 3 4 5 6 7 8
<πραγματικός>	::=	<θετικός πραγματικός> - <θετικός πραγματικός> <θετικός ακέραιος> . <θετικός ακέραιος> e (+ -) <θετικός ακέραιος> - <θετικός ακέραιος> . <θετικός ακέραιος> e (+ -) <θετικός ακέραιος>
<JSON array>	::=	[(<πραγματικός> <αλφαριθμητικό> ,) ⁺ <πραγματικός> <αλφαριθμητικό>]
<θετικός ακέραιος 0 ή 1>	::=	(0 1)

BNF Ερωτήματος 2 (μόνο τις διαφορές από την 1)

<content>	::=	"content": [<gameid> , <drawID> , <drawTime> , <status> , <drawBreaks> , <visualDraw> , <pricePoints> , <winningNumbers> , <prizeCategories> , <wagerStatistics>]
<totalPages>	::=	"totalPages": <θετικός ακέραιος>
<totalElements>	::=	"totalElements": <θετικός ακέραιος>
<last>	::=	"last": <boolean>
<numberOfElements>	::=	"numberOfElements": <θετικός ακέραιος>
<sort>	::=	"sort": [<direction> , <property> , <ignoreCase> , <nullHandling> , <descending> , <ascending>]
<direction>	::=	"direction": <αλφαριθμητικό>
<property>	::=	"property" : <αλφαριθμητικό> . <αλφαριθμητικό>
<ignoreCase>	::=	"ignoreCase": <boolean>
<nullHandling>	::=	"nullHandling": <αλφαριθμητικό>
<descending>	::=	"descending": <boolean>
<ascending>	::=	"ascending": <boolean>
<first>	::=	"first": <boolean>
<size>	::=	"size": <θετικός ακέραιος>
<number>	::=	"number": <θετικός ακέραιος>
<boolean>	::=	true false

Σημειώσεις:

Τα παραπάνω BNF δεν αναπαριστούν case-sensitive τους κανόνες που ορίσαμε στη Flex και στη Bison, αλλά αποτελούν μια απόπειρα για πιο εύκολα κατανοητή αναπαράσταση των κανόνων που ορίσαμε.

Bison.y

```
%{  
    #include <stdio.h>  
    void yyerror(const char *);  
    extern FILE *yyin;  
    extern FILE *yyout;  
    extern int yylineno;  
%}
```

```
%define parse.error verbose  
%locations
```

```
//Symbol Tokens
```

```
%token '.'  
%token ','  
%token '{'  
%token '}'  
%token '['  
%token ']'  
%token '-'  
%token '+'
```

```
//Identifier Tokens
```

```
%token LAST  
%token ACTIVE  
%token GAMEID  
%token DRAWID  
%token DRAW_TIME  
%token STATUS
```

%token DRAW_BREAK
%token VISUAL_DRAW
%token PRICE_POINTS
%token AMOUNT
%token WINNING_NUMBERS
%token LIST
%token BONUS
%token PRIZE_CATEGORIES
%token ID
%token DIVIDENT
%token WINNERS
%token DISTRIBUTED
%token JACKPOT
%token FIXED
%token CATEGORY_TYPE
%token GAMETYPE
%token MINIMUM_DISTRIBUTED
%token WAGER_STATISTICS
%token COLUMNS
%token WAGERS
%token ADDON

//Extra exc2 Tokens

%token CONTENT
%token TOTAL_PAGES
%token TOTAL_ELEMENTS
%token NUMBER_OF_ELEMENTS
%token SORT
%token DIRECTION
%token PROPERTY

%token IGNORE_CASE
%token NULL_HANDLING
%token DESCENDING
%token ASCENDING
%token FIRST
%token SIZE
%token NUMBER
%token IDONE
%token CLASSITEM

//Value Tokens

%token POSITIVE_INTEGER
%token ALPHANUMERIC
%token POSITIVE_INTEGER_TWO_TO_EIGHT
%token POSITIVE_INTEGER_ZERO_OR_ONE
%token TRUE
%token FALSE

%%

```
compiler: '{' last ',' active '}' {fprintf(yyout, "\nThe Syntax was Correct!\n");}  
        | '{' content ',' totalPages ',' totalElements ',' last2 ',' numberOfElements  
' ,' sort ',' first ',' size ',' number '}' {fprintf(yyout, "\nThe Syntax was Correct!\n");}  
        ;
```

last: LAST '{' gameId ',' drawId ',' drawTime ',' status ',' drawBreak ',' visualDraw ','
pricePoints ',' winningNumbers ',' prizeCategories ',' wagerStatistics '}' {}

active: ACTIVE '{' gameId ',' drawId ',' drawTime ',' status ',' drawBreak ','
visualDraw ',' pricePoints ',' prizeCategories ',' wagerStatistics '}' {}

positivereal: POSITIVE_INTEGER '.' POSITIVE_INTEGER {}
real: positivereal | '-' positivereal {}
arrayint: POSITIVE_INTEGER ',' arrayint | POSITIVE_INTEGER {}
arrayposints: '[' arrayint ']' {}
arrayposint: '[' POSITIVE_INTEGER ']' {}
jsoncontent: ALPHANUMERIC ',' jsoncontent | ALPHANUMERIC {}
jsonarray: '[' jsoncontent ']' | '[' ']' {}

gameId: GAMEID POSITIVE_INTEGER {}
drawId: DRAWID POSITIVE_INTEGER {}
drawTime: DRAW_TIME POSITIVE_INTEGER {}
status: STATUS ALPHANUMERIC {}
drawBreak: DRAW_BREAK POSITIVE_INTEGER {}
visualDraw: VISUAL_DRAW POSITIVE_INTEGER {}
pricePoints: PRICE_POINTS '{' amount '}' {}
amount: AMOUNT positivereal {}
winningNumbers: WINNING_NUMBERS '{' list ',' bonus '}' {}
list: LIST arrayposints{}
bonus: BONUS arrayposint{}
prizecontent: '{' idone ',' dividant ',' winners ',' distributed ',' jackpot ',' fixed ','
categoryType ',' gameType ',' minimumDistributed '}'
| '{' id ',' dividant ',' winners ',' distributed ',' jackpot ',' fixed ','
categoryType ',' gameType '}' {}
prizelisting: prizecontent ',' prizelisting | prizecontent
prizeCategories: PRIZE_CATEGORIES '[' prizelisting ']' {}
idone: IDONE {}
id: ID POSITIVE_INTEGER_TWO_TO_EIGHT {}
divident: DIVIDENT positivereal {}
winners: WINNERS POSITIVE_INTEGER {}
distributed: DISTRIBUTED positivereal {}

jackpot: JACKPOT positivereal {}
fixed: FIXED positivereal {}
categoryType: CATEGORY_TYPE POSITIVE_INTEGER_ZERO_OR_ONE {}
gameType: GAMETYPE ALPHANUMERIC {}
minimumDistributed: MINIMUM_DISTRIBUTED positivereal
wagerStatistics: WAGER_STATISTICS '{' columns ',' wagers ',' addOn '}' {}
columns: COLUMNS POSITIVE_INTEGER {}
wagers: WAGERS POSITIVE_INTEGER {}
addOn: ADDON jsonarray {}

//Extra exc2 Tokens

contentlisting: '{' gameId ',' drawId ',' drawTime ',' status ',' drawBreak ','
visualDraw ',' pricePoints ',' winningNumbers ',' prizeCategories ',' wagerStatistics
'}' ',' contentlisting | '{' gameId ',' drawId ',' drawTime ',' status ',' drawBreak ','
visualDraw ',' pricePoints ',' winningNumbers ',' prizeCategories ',' wagerStatistics
'}' ',' {};
content: CONTENT '[' contentlisting ']' {}

sort: SORT '[' '{' direction ',' property ',' ignoreCase ',' nullHandling ',' descending ','
ascending '}' ']' {};

totalPages: TOTAL_PAGES POSITIVE_INTEGER {}
totalElements: TOTAL_ELEMENTS POSITIVE_INTEGER {}
boolean: TRUE | FALSE {}
last2: LAST boolean{}
numberOfElements: NUMBER_OF_ELEMENTS POSITIVE_INTEGER {}
direction: DIRECTION ALPHANUMERIC {}
property: PROPERTY CLASSITEM{}
ignoreCase: IGNORE_CASE boolean {}
nullHandling: NULL_HANDLING ALPHANUMERIC {}

descending: DESCENDING boolean {}
ascending: ASCENDING boolean {}
first: FIRST boolean {}
size: SIZE POSITIVE_INTEGER {}
number: NUMBER POSITIVE_INTEGER {}

%%

```
void yyerror(const char *errmsg)
{
    fprintf(yyout, "\nError in line: %d\n%s\n", yylineno, errmsg);
}
```

```
int main (int argc, char **argv)
{
    ++argv; --argc;
    if (argc > 0)
        yyin = fopen(argv[0], "r");
    else
        yyin = stdin;
    yyout = fopen("output", "w");
    yyparse();
    return 0;
}
```

Flex.l

```
%{  
    #include <stdio.h>  
    #include "y.tab.h"  
    int list_count=0;  
    int prizeCategoriesCounter = 0;  
    extern void yyerror(const char *);  
    #define PRINT fwrite (yytext, yyleng, 1, yyout)  
%}
```

```
%s ALPHA BETA GAMMA CHECK LISTCHECK
```

```
%option noyywrap yylineno
```

```
digit [0-9]
```

```
num      {digit}+
```

```
plus     "+"
```

```
minus    "-"
```

```
true     "true"
```

```
false    "false"
```

```
zeroone [01]
```

```
twotoeight [2-8]
```

```
alphanumeric [a-zA-Z0-9]
```

```
classitem {alphanumeric}\.{alphanumeric}
```

```
%%
```

```
<INITIAL>
```

```
{
```

```
"\n"      {}
```

```
"\-"      { PRINT; return '-'; }
```

```
"\+"      { PRINT; return '+'; }
```

```

"\,"      { PRINT; return ','; }
"\{"      { PRINT; return '{'; }
"\}"      { PRINT; return '>'; }
"\["      { PRINT; return '['; }
"\]"      { PRINT; return ']'; }
"\."      { PRINT; return '.'; }
"\id\": 1" { PRINT; return IDONE; }
"\id\":1"  { PRINT; return IDONE; }
"\last\":" { PRINT; return LAST; }
"\active\":" { PRINT; return ACTIVE; }
"\gameId\":" { BEGIN(CHECK); PRINT; return GAMEID; }
"\drawId\":" { PRINT; return DRAWID; }
"\drawTime\":" { PRINT; return DRAW_TIME; }
"\status\":" { BEGIN(ALPHA); PRINT; return STATUS; }
"\drawBreak\":" { PRINT; return DRAW_BREAK; }
"\visualDraw\":" { PRINT; return VISUAL_DRAW; }
"\pricePoints\":" { PRINT; return PRICE_POINTS; }
"\amount\":" { PRINT; return AMOUNT; }
"\winningNumbers\":" { PRINT; return WINNING_NUMBERS; }
"\list\":" { BEGIN(LISTCHECK); PRINT; return LIST; }
"\bonus\":" { PRINT; return BONUS; }
"\prizeCategories\":" { prizeCategoriesCounter = 0; PRINT; return
PRIZE_CATEGORIES; }
"\id\":" { BEGIN(BETA); PRINT; return ID; }
"\divident\":" { prizeCategoriesCounter++; checkMaximumPrizeCategories();
PRINT; return DIVIDENT; }
"\winners\":" { PRINT; return WINNERS; }
"\distributed\":" { PRINT; return DISTRIBUTED; }
"\jackpot\":" { PRINT; return JACKPOT; }
"\fixed\":" { PRINT; return FIXED; }

```

```

"\categoryType\":" { BEGIN(BETA); PRINT; return CATEGORY_TYPE; }
"\gameType\":" { BEGIN(ALPHA); PRINT; return GAMETYPE; }
"\minimumDistributed\":" { PRINT; return MINIMUM_DISTRIBUTED; }
"\wagerStatistics\":" { checkMinimumPrizeCategories(); PRINT; return
WAGER_STATISTICS; }
"\columns\":" { PRINT; return COLUMNS; }
"\wagers\":" { PRINT; return WAGERS; }
"\addOn\":" { BEGIN(ALPHA); PRINT; return ADDON; }
{num} { yylval = atoi(yytext); PRINT; return POSITIVE_INTEGER; }
"\content\":" { PRINT; return CONTENT;}
"\sort\":" { PRINT; return SORT;}
"\totalPages\":" { PRINT; return TOTAL_PAGES;}
"\totalElements\":" { PRINT; return TOTAL_ELEMENTS;}
"\last2\":" { PRINT; return LAST;}
"\numberOfElements\":" { PRINT; return NUMBER_OF_ELEMENTS;}
"\direction\":" { BEGIN(ALPHA); PRINT; return DIRECTION;}
"\property\":" { BEGIN(GAMMA); PRINT; return PROPERTY;}
"\ignoreCase\":" { PRINT; return IGNORE_CASE;}
"\nullHandling\":" { BEGIN(ALPHA); PRINT; return NULL_HANDLING;}
"\descending\":" { PRINT; return DESCENDING;}
"\ascending\":" { PRINT; return ASCENDING;}
"\first\":" { PRINT; return FIRST;}
"\size\":" { PRINT; return SIZE;}
"\number\":" { PRINT; return NUMBER;}
"true" { PRINT; return TRUE; }
"false" { PRINT; return FALSE; }
}

```

<ALPHA>

```
{
```

```

        "\["      { PRINT; return '['; }
        "\]"      { BEGIN(INITIAL); PRINT; return ']'; }
        {alphanumeric} { BEGIN(INITIAL); PRINT; return ALPHANUMERIC; }
    }

```

<BETA>

```

{
    {zeroone}    { BEGIN(INITIAL); PRINT; return
POSITIVE_INTEGER_ZERO_OR_ONE; }
    {twotoeight} { BEGIN(INITIAL); PRINT; return
POSITIVE_INTEGER_TWO_TO_EIGHT;}
}

```

<GAMMA>

```

{
    {classitem}  { BEGIN(INITIAL); PRINT; return CLASSITEM; }
}

```

<CHECK>{

```

"\n"      {}
{num}     { yyval = atoi(yytext); check(yyval); BEGIN(INITIAL); PRINT; return
POSITIVE_INTEGER;}

}

```

<LISTCHECK>{

```

"\n"      {}
"\\"      { PRINT; return ','; }

```

```

{num}      {yyval = atoi(yytext); list_count++; num_range(yyval); PRINT; return
POSITIVE_INTEGER;}
"\["      { PRINT; return '['; }
"\]"      { list_check(list_count);list_count=0; BEGIN(INITIAL); PRINT; return ']';
}

```

```

}

```

```

%%

```

```

void num_range(int val){
    if((0<val)&&(val<45))
        return;
    fprintf(yyout,"\nError: In List Item There Is Value (%d) Which Is Outside
Of The Allowed Range!\n",val);
    exit(0);
}

```

```

void list_check(int count){
    if(count==5)
        return;
    fprintf(yyout, "\nError: List Embeded JSON Items Are  %d, Not
5!\n",count);
    exit(0);
}

```

```

void check (int val){
    if(val == 1100)
        return;
    else if(val == 1110)
        return;
    else if(val == 2100)

```

```

        return;
    else if(val == 2101)
        return;
    else if(val == 5103)
        return;

    else if(val == 5104)
        return;
    else if(val == 5106)
        return;

    fprintf(yyout, "\nError: Invalid Gameld!\n");
    exit(0);
}

void checkMaximumPrizeCategories()
{
    if(prizeCategoriesCounter > 8)
    {
        fprintf(yyout, "\nError: prizeCategories Has More Than 8 Embedded
JSON Objects!\n");
        exit(0);
    }
    return;
}

void checkMinimumPrizeCategories()
{
    if(prizeCategoriesCounter < 8)
    {

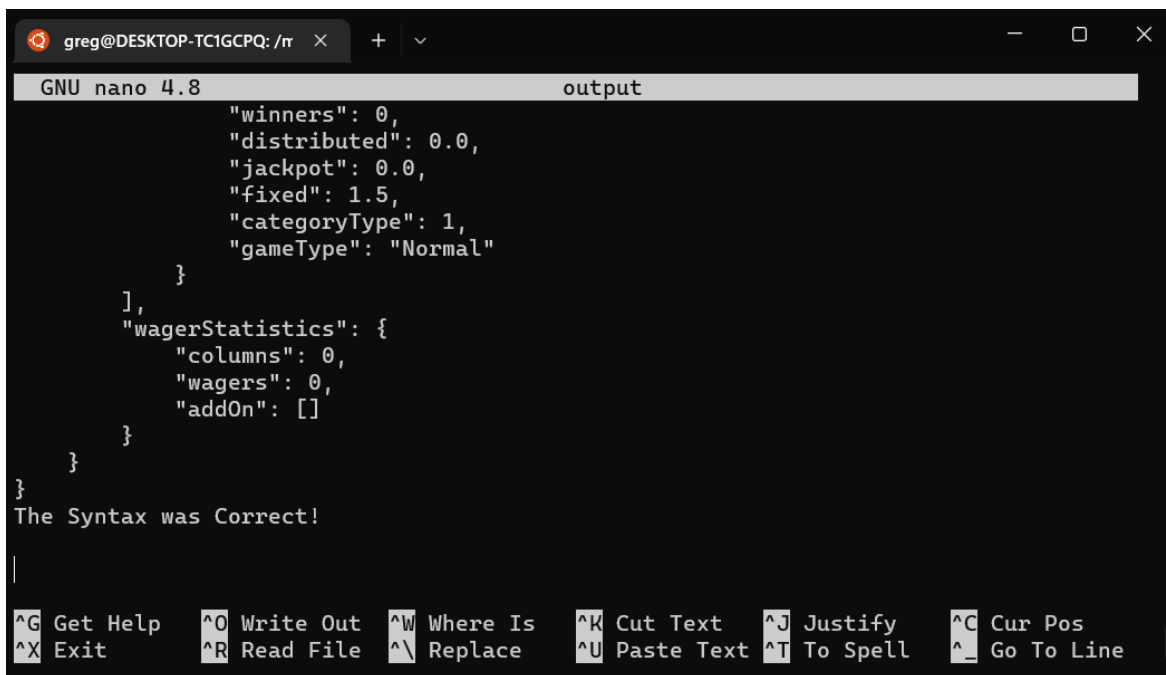
```



```
        fprintf(yyout, "\\nError: prizeCategories Has Less Than 8 Embedded  
JSON Objects!\\n");  
        exit(0);  
    }  
    return;  
}
```

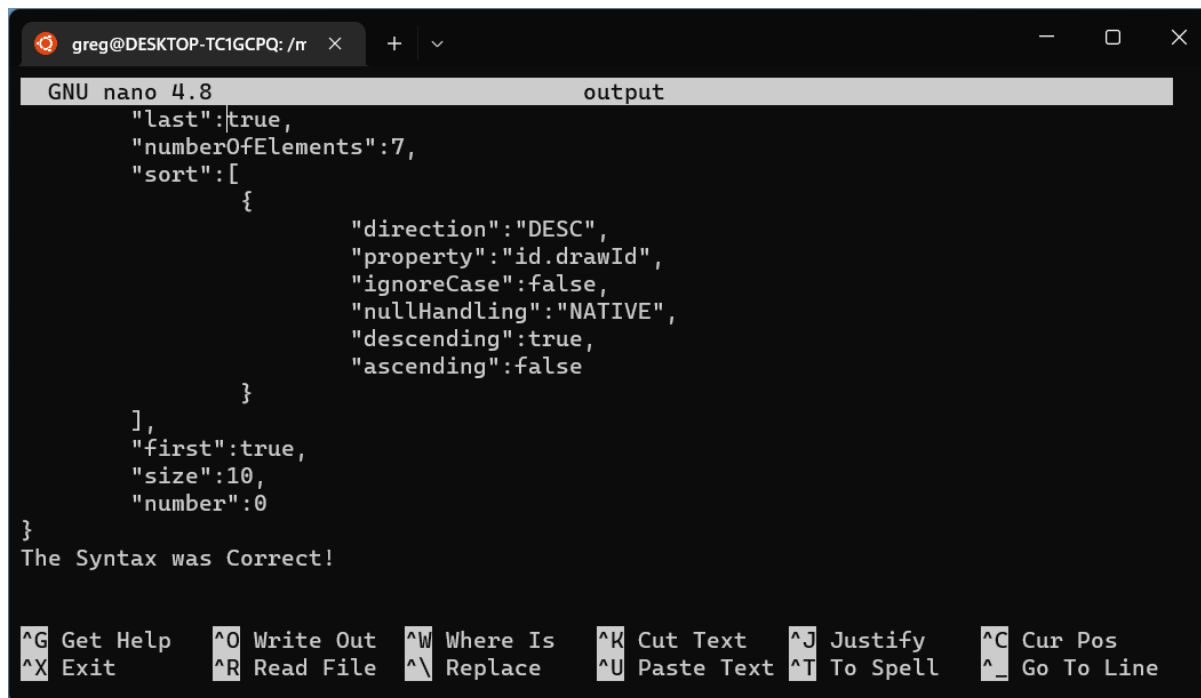
Screenshots:

1. Επιτυχία last_result.json:



```
greg@DESKTOP-TC1GCPQ: /r x + v  
GNU nano 4.8 output  
    "winners": 0,  
    "distributed": 0.0,  
    "jackpot": 0.0,  
    "fixed": 1.5,  
    "categoryType": 1,  
    "gameType": "Normal"  
  }  
],  
  "wagerStatistics": {  
    "columns": 0,  
    "wagers": 0,  
    "addOn": []  
  }  
}  
The Syntax was Correct!  
|  
^G Get Help  ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos  
^X Exit      ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

2. Επιτυχία range_result.json

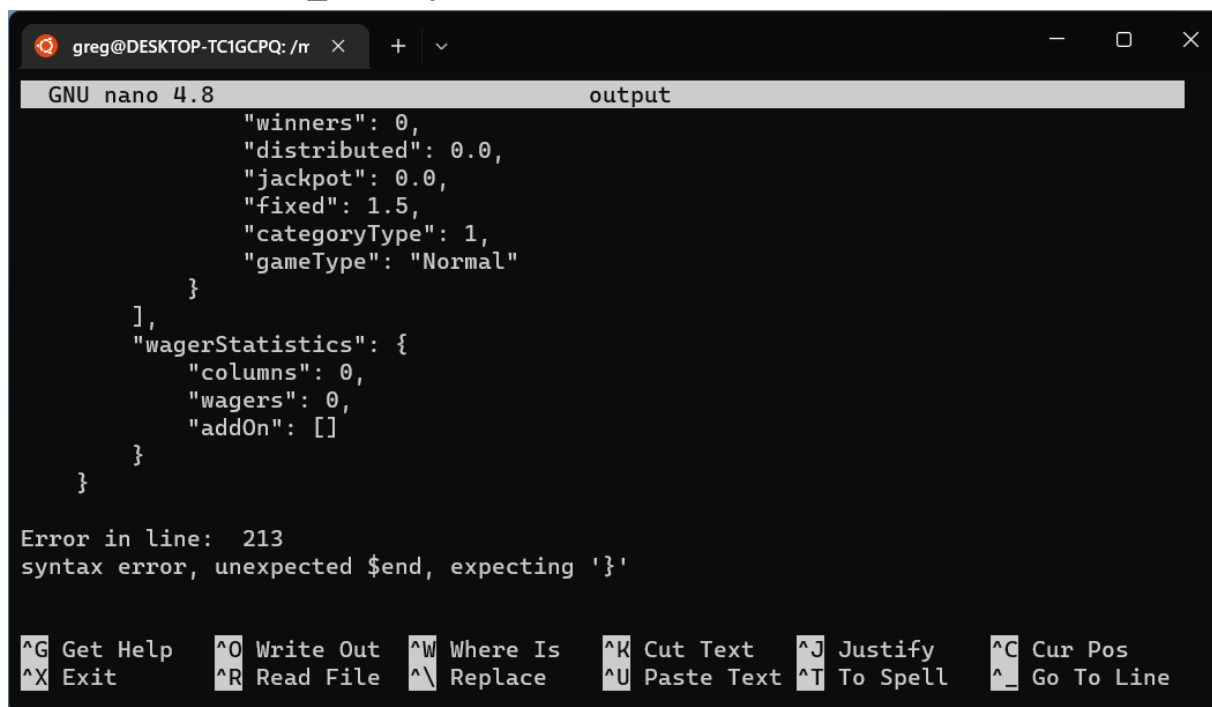


A terminal window titled 'greg@DESKTOP-TC1GCPQ: /r' shows the GNU nano 4.8 editor editing a file named 'output'. The JSON content is as follows:

```
"last":true,
"numberOfElements":7,
"sort":[
  {
    "direction":"DESC",
    "property":"id.drawId",
    "ignoreCase":false,
    "nullHandling":"NATIVE",
    "descending":true,
    "ascending":false
  }
],
"first":true,
"size":10,
"number":0
}
```

The message 'The Syntax was Correct!' is displayed at the bottom of the editor. The nano editor's command shortcuts are visible at the bottom of the terminal window.

3. Μη επιτυχία last_result.json:




A terminal window titled 'greg@DESKTOP-TC1GCPQ: /r' shows the GNU nano 4.8 editor editing a file named 'output'. The JSON content is as follows:

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

An error message is displayed at the bottom of the editor: 'Error in line: 213 syntax error, unexpected \$end, expecting '''''. The nano editor's command shortcuts are visible at the bottom of the terminal window.

4. Μη επιτυχή range_result.json:

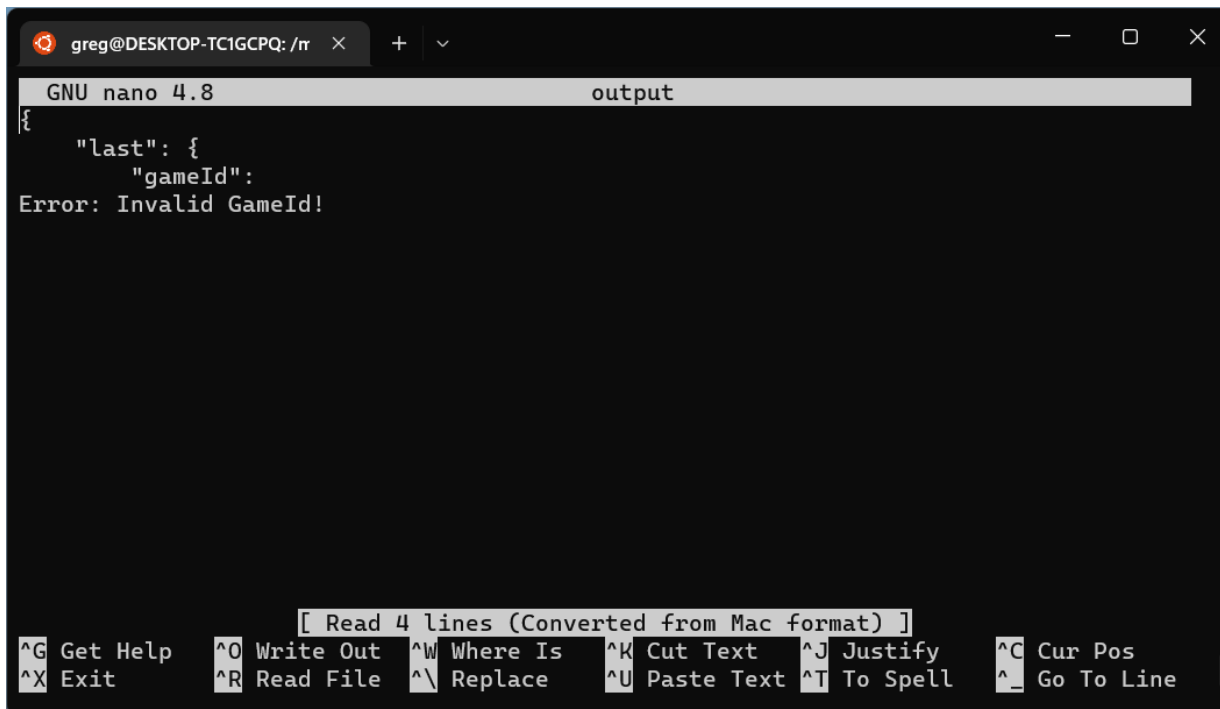


```
greg@DESKTOP-TC1GCPQ: /r x + v
GNU nano 4.8 output
{
  "last":true,
  "numberOfElements":7,
  "sort":[
    {
      "direction":"DESC",
      "property":"id.drawId",
      "ignoreCase":false,
      "nullHandling":"NATIVE",
      "descending":true,
      "ascending":false
    }
  ],
  "first":true,
  "size":10,
  "number":0
}

Error in line: 430
syntax error, unexpected $end, expecting '}'

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

5. Μη ορθό “gameId”:

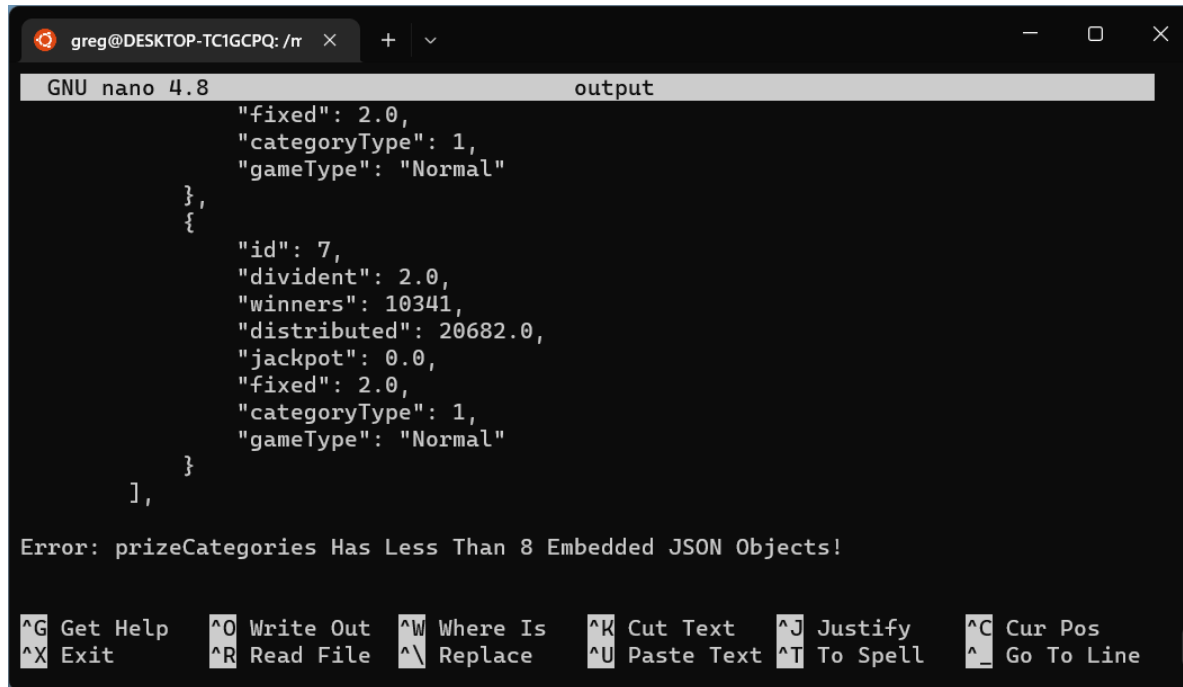


```
greg@DESKTOP-TC1GCPQ: /r x + v
GNU nano 4.8 output
{
  "last": {
    "gameId":
Error: Invalid gameId!

[ Read 4 lines (Converted from Mac format) ]

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line
```

6. prizeContents JSON αριθμός < 7

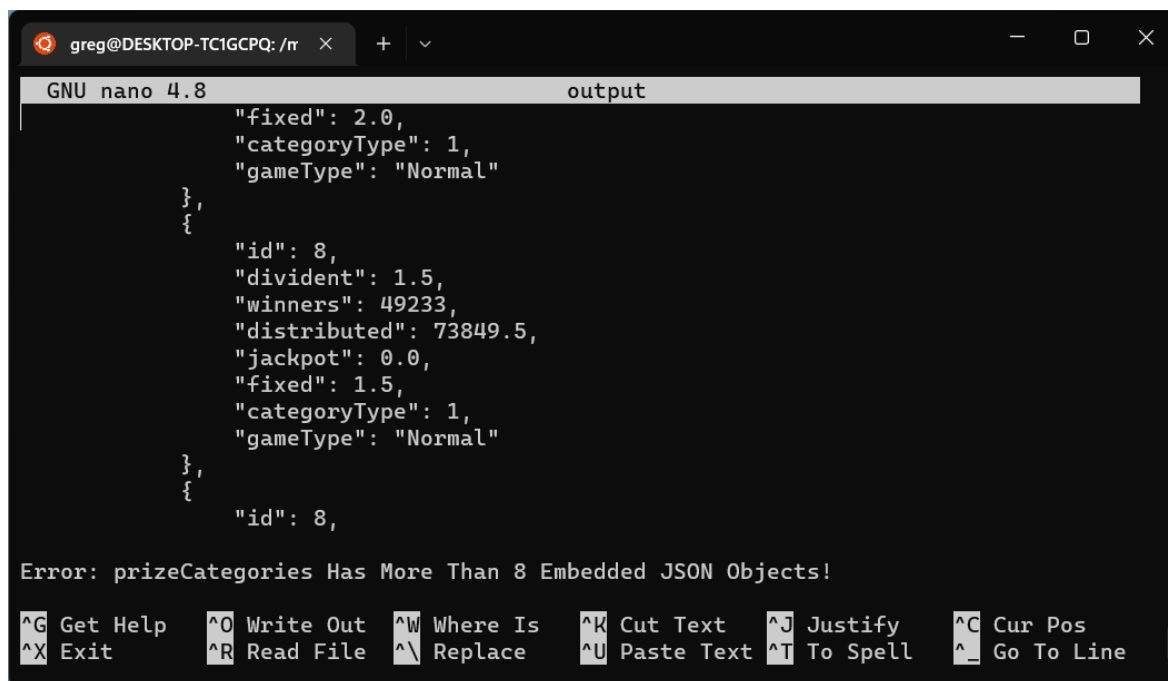


The screenshot shows a terminal window with the title bar 'greg@DESKTOP-TC1GCPQ: /r'. The editor is GNU nano 4.8, editing a file named 'output'. The JSON content is as follows:

```
{
  "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"
}
],
```

Below the JSON, an error message is displayed: "Error: prizeCategories Has Less Than 8 Embedded JSON Objects!". At the bottom, a status bar shows various keyboard shortcuts: ^G Get Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\ Replace, ^K Cut Text, ^U Paste Text, ^J Justify, ^T To Spell, ^C Cur Pos, and ^_ Go To Line.

7. prizeContents JSON αριθμός > 7:

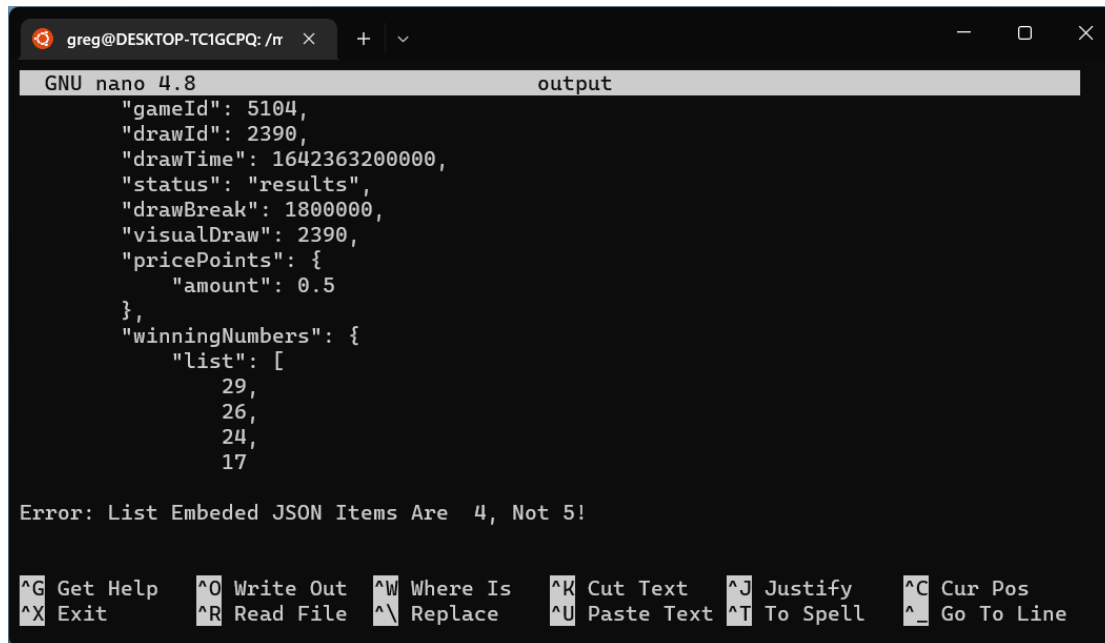


The screenshot shows a terminal window with the title bar 'greg@DESKTOP-TC1GCPQ: /r'. The editor is GNU nano 4.8, editing a file named 'output'. The JSON content is as follows:

```
{
  "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"
},
{
  "id": 8,
```

Below the JSON, an error message is displayed: "Error: prizeCategories Has More Than 8 Embedded JSON Objects!". At the bottom, a status bar shows various keyboard shortcuts: ^G Get Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^\ Replace, ^K Cut Text, ^U Paste Text, ^J Justify, ^T To Spell, ^C Cur Pos, and ^_ Go To Line.

8. List doesn't have 5 integers

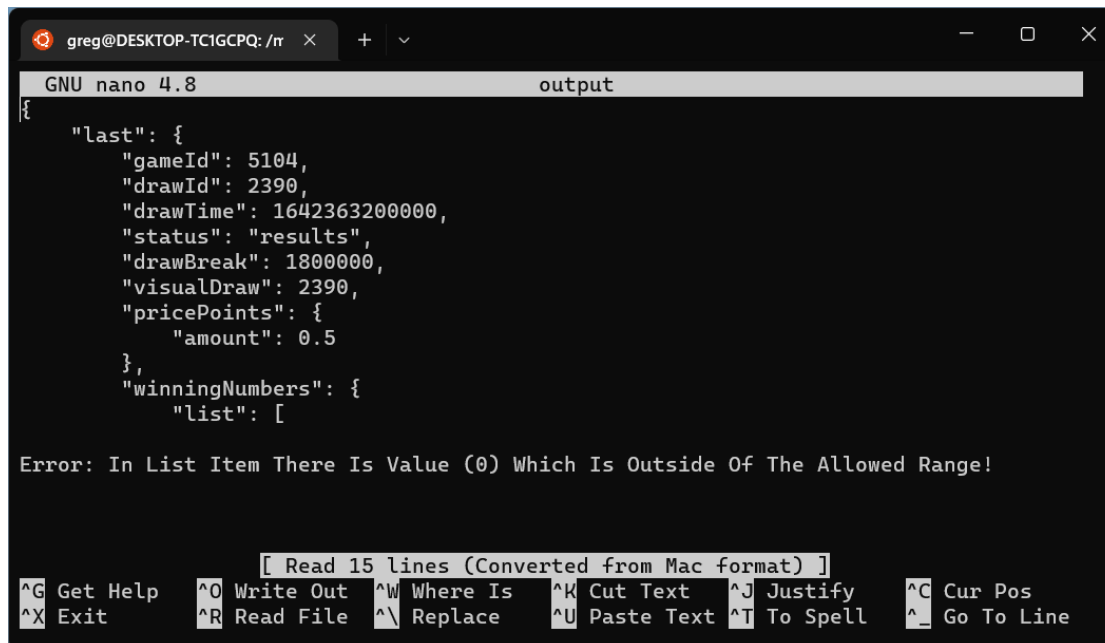


The screenshot shows a terminal window with the title bar "greg@DESKTOP-TC1GCPQ: /r". The editor is GNU nano 4.8, editing a file named "output". The JSON content is as follows:

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

Below the JSON, an error message is displayed: "Error: List Embedded JSON Items Are 4, Not 5!". At the bottom of the terminal, a row of keyboard shortcuts is visible: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^_ Replace, ^U Paste Text, ^T To Spell, and ^_ Go To Line.

9. List has integers not in [1,45] range:



The screenshot shows a terminal window with the title bar "greg@DESKTOP-TC1GCPQ: /r". The editor is GNU nano 4.8, editing a file named "output". The JSON content is as follows:

```
{
  "last": {
    "gameId": 5104,
    "drawId": 2390,
    "drawTime": 1642363200000,
    "status": "results",
    "drawBreak": 1800000,
    "visualDraw": 2390,
    "pricePoints": {
      "amount": 0.5
    },
    "winningNumbers": {
      "list": [

```

Below the JSON, an error message is displayed: "Error: In List Item There Is Value (0) Which Is Outside Of The Allowed Range!". A status bar at the bottom of the editor indicates "[Read 15 lines (Converted from Mac format)]". The same row of keyboard shortcuts from the previous screenshot is visible at the bottom of the terminal.