

بنام خدا



اصول طراحی کامپایلر

تمرین سوم

```
grammar Q1;
// Lexer rules
WS : [ \t\r\n ]+ -> skip;
INT : 'int';
BOOLEAN : 'boolean';
IF : 'if';
WHILE : 'while';
TRUE : 'true';
FALSE : 'false';
ID : [a-zA-Z]+;
NUMBER : [0-9]+;
ASSIGN : '=';
SEMI : ';';
LPAREN : '(';
RPAREN : ')';
LBRACE : '{';
RBRACE : '}';
ADD : '+';
SUB : '-';
EQ : '==';
GT : '>';
LT : '<';
COMMENT : '#' ~[\r\n]* -> skip;
// Parser rules
program : statement+;
statement : variableDeclaration
          | assignment
          | ifStatement
          | whileStatement
          | COMMENT;
variableDeclaration : type ID (ASSIGN expression)? SEMI;
type : INT | BOOLEAN;
assignment : ID ASSIGN expression SEMI;
expression : NUMBER
           | TRUE
           | FALSE
           | ID
           | expression ADD expression
           | expression SUB expression
           | expression EQ expression
           | expression GT expression
           | expression LT expression
           | LPAREN expression RPAREN;
ifStatement : IF LPAREN expression RPAREN LBRACE statement RBRACE;
whileStatement : WHILE LPAREN expression RPAREN LBRACE statement RBRACE;
```

سوال ۱.

در این سوال قصد داریم یک زبان ساده برنامه نویسی به نام زبان چالوس-اوتاق را طراحی کنیم. این زبان شامل دستورات زیر است:

1. انواع جملات تخصیصی (Assignment)، شرطی (if)، حلقه (while)، جملات ترکیبی (Compound)
2. تعریف با declaration نوع متغیرها شامل int و boolean.
3. قوانین لغوی شامل تعریف اعداد، کامنت، عملگرهای محاسباتی، عملگرهای مقایسه ای، عملگرهای منطقی و مقادیر true و false

- تعریف و مقدار دهی متغیرها (این زبان تنها از متغیرهای عدد صحیح و بولین پشتیبانی می‌کند)

```
int x;  
x=10;  
boolean b;  
b=true;  
int y=76;  
boolean c=false;
```

- وجود عملگرهای مقایسه ای بزرگتر، مساوی و کوچکتر در زبان برای متغیرهای عدد صحیح. خروجی این عملگرها یک boolean است.
- وجود عملگرهای محاسبه گر جمع و تفریق برای اعداد صحیح.
- وجود دستور شرطی تنها به شکل زیر:

```
if(boolean condition) {  
    State;  
}
```

- وجود دستور برای ایجاد حلقه تنها به شکل زیر:

```
while(boolean condition) {  
    State;  
}
```

- وجود کامنت تنها به صورت زیر:

```
#comment
```

یک گرامر با فرمت g4 برای زبان چالوس-اوتاق طراحی نمایید.

سوال ۲.

با کمک ابزار Antlr و زبان پایتون برنامه ای بنویسید که یک رشته حاوی یک فایل دستورات زبان چالوس-اوتاق را در ورودی گرفته، در ادامه تغییراتی را روی آن اعمال کند. این تغییرات بدین شکل است که در ابتدای هر کامنت نام خانوادگی شما و در انتهای هر کامنت شماره دانشجویی شما را نوشته و کامنت بدست آمده را جایگزین کامنت قبلی کند.

```
#comment
```

میشود:

```
#<your_lastname>comment<your_student_id>
```

سوال ۳.

با استفاده از زبان پایتون و ابزار انتلر برنامه ای بنویسید که یک رشته حاوی دستورات زبان چالوس-اوتاق را دریافت کرده و بیشترین عمق دستورات را در خروجی چاپ کند. هر موقع وارد بدنه یک دستور شرطی یا یک حلقه میشویم، عمق کد یک درجه افزایش یافته و وقتی از هر کدام خارج می‌شویم عمق کد یک درجه افزایش میابد. به عنوان مثال حداکثر عمق قطعه کد زیر برابر ۲ است.

```

0
if(condition) {
    State; 1
}
0
while(condition) {
    1 if(condition) {
        State; 2
    }
}
0

```

کامپس

سوال ۴.

گرامر زیر داده شده است. با توجه به این گرامر به سوالات زیر پاسخ دهید.

$$e \rightarrow t | e + t | e - t$$

$$t \rightarrow f | t * f | t / f$$

$$f \rightarrow (e) | n$$

?

- آیا گرامر فوق LL1 است؟ در صورت LL1 بودن آنرا اثبات کنید و در صورت نبودن آنرا به صورت LL1 درآورید.
- با نوشتن مجموعه های first و follow جدول تجزیه گرامر فوق را رسم کنید.
- با استفاده از یک Stack رشته $n*n-n*n$ را تجزیه کنید.
- درخت تجزیه رشته داده شده را رسم کنید.