

La primul test cu specificatiile d-voastra pentru Lex si Yacc, am obtinut asa ceva:

```
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
This is a university
Verb phrase
Sentence
```

1. Observatii in privinta specificatiei pentru Yacc

Sectiunea cu regulile gramaticale (productions) ati scris-o in felul urmator:

```
sentence : SUBJ verb_phrase object {printf("Sentence\n");}

verb_phrase : ADVB VERB | VERB {printf("Verb phrase\n");}

object : ARTC NOUN | NOUN {printf("Object\n");}
```

In felul in care ati scris regulile pentru simbolurile neterminale `verb_phrase` si `object` nu veti putea obtine mesaje de ‘trace’ decit pentru partea a 2-a a fiecarui set de reguli!!!

Eu v-as sugera sa scrieti asa:

```
sentence : SUBJ verb_phrase object      {printf("Sentence\n");}
          ;

verb_phrase : ADVB VERB                  { printf("Verb phrase: ADVB VERB\n"); }
            | VERB                       {printf("Verb phrase: VERB\n"); }
          ;

object : ARTC NOUN                       { printf("Object: ARTC NOUN\n"); }
        | NOUN                           {printf("Object: NOUN\n"); }
      ;
```

Cu alte cuvinte sa faceti “trace” prin intermediul “actiunilor semantice” pentru toate regulile gramaticale, corespunzatoare unui simbol neterminal.

In plus, la sfirsitul fiecarui set de reguli gramaticale pentru un acelasi simbol terminal in LHS ar trebui sa puneti ‘;’ prin aceasta semnalind sfirsitul setului de reguli.

2. Observatii in privinta specificatiei pentru Lex

Ati omis includerea regulilor pentru:

- identificarea (si consumarea) spatiilor si TAB-urilor:
[\t]+ ;
 - identificarea (si raportarea spre analizorul sintactic) NEW LINE-ului:
[\n] { return '\0'; }
- In acest fel semnalati analizorului sintactic sfirsitul fluxului de intrare!
- detectarea si raportarea caracterelor nedorite:
. { printf("Illegal character: %c\n", yytext[0]); }

Cu aceste modificari iata ce am obtinut la rularea analizorului sintactic:

```
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
This is a university
Verb phrase: VERB
Object: ARTC NOUN
Sentence
antonc@ubuntu-16:~/FLT/L03$
```

Pentru ca lucrurile sa fie si mai elocvente ati putea adauga instructiuni de ‘trace’ si la nivelul specificatiei Lex si atunci veti obtine ceva de genul:

```
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
This is a university
Lex - SUBJ: This
```

```
Lex - VERB: is
Verb phrase: VERB
Lex - ARTC: a
Lex - NOUN: university
Object: ARTC NOUN
Sentence
xxxx@ubuntu-16:~/FLT/L03$
```

Daca in specificatia pentru Lex ati mai adauga doua cuvinte pentru simbolul terminal NOUN (girl si program) am putea executa si urmatoarele teste:

```
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
I am a girl
Lex - SUBJ: I
Lex - VERB: am
Verb phrase: VERB
Lex - ARTC: a
Lex - NOUN: girl
Object: ARTC NOUN
Sentence
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
this computer run program
Illegal character: t
Illegal character: h
Lex - VERB: is
syntax error
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
This computer run program
Lex - SUBJ: This
Lex - SUBJ: computer
syntax error
xxxx@ubuntu-16:~/FLT/L03$ ./phrase
computer run program
Lex - SUBJ: computer
Lex - VERB: run
Verb phrase: VERB
Lex - NOUN: program
Object: NOUN
Sentence
xxxx@ubuntu-16:~/FLT/L03$
```

Sunt convins ca veti putea sa gasiti explicatiile necesare pentru erorile semnalate de analizorul sintactic, respectiv cel lexical!