1. MiniC: Scanner und Parser mit (f)lex und yacc/bison

MiniC.l

```
/*MiniC.l:
  ------
  Description of the lexical structure for MiniCw.
=====================================================================*/

%{

  #include "MiniC.tab.h" /*generated by yacc/bison from MiniC.y
                           if option -d is used, defines NUMBER      */



%}

%%

[ \t\r\n]+   { ; }          /*ignore white space: blanks, tabs and new line
*/

[0-9]+   { return NUMBER; }

void                     { return VOID; }
main                     { return MAIN; }
int                      { return INT; }
scanf                    { return SCANF; }
printf                   { return PRINTF; }

[A-Za-z_][A-Za-z0-9\_]* { return IDENT; }

.                        { return yytext[0]; } /*return all other chars
                                as tokens: '+', '-', ...           */
%%

int yywrap() {
  return 1; /*on end of input: no further files to scan             */
} /*yywrap*/

/* End of MiniC.l
=====================================================================*/
```

MiniC.y

```
/*MiniC.y:
  ------
  Attributed grammar for MiniC.
```

```
====================================================================*/

%{
    #include <stdio.h>

    extern
%}

%token NUMBER
%token IDENT
%token VOID
%token MAIN
%token INT
%token SCANF
%token PRINTF

%%

MiniC: VOID MAIN '(' ')' '{'
                    OptVarDecl
                    StatSeq
                    '}'
  ;

OptVarDecl: /* eps */
  | VarDecl
  ;

VarDecl: INT IdList ';'
  ;

IdList: IDENT
  | IdList ',' IDENT
  ;

StatSeq: Stat
  | StatSeq Stat
  ;

Stat: ';'
  | IDENT '=' Expr ';'
  | SCANF '(' IDENT ')' ';'
  | PRINTF '(' Expr ')' ';'
  ;

Expr: Term
  | Expr '+' Term
  | Expr '-' Term
  ;

Term: Fact
  | Term '*' Fact
  | Term '/' Fact
  ;
```

```
Fact: IDENT
    | NUMBER
    | '(' Expr ')'
    ;

%%
extern int yylineno;

int yyerror(char *msg) {
  printf("error: %s in line %d\n", msg, yylineno);
  return 0;
} /*yyerror*/

int main(int argc, char *argv[]) {
  yyparse();
  return 0;
} /*main*/


/* End of Calc.y
====================================================================*/
```

Commands

```
..\Flex-2.5.37\flex.exe --yylineno MiniC.l
..\Bison-2.7\bison.exe -g -d MiniC.y
gcc lex.yy.c MiniC.tab.c -o MiniC.exe
MiniC.exe < SVP.mc
```

SVP.mc

```
void main() {
  int a, b, cs;
  scanf(a);
  scanf(b);
  cs = (a * a) + (b * b);
  printf(cs);
}
```

2. MiniCpp: Scanner und Parser mit (f)lex und yacc/bison UND ...