%{

#include <stdio.h>

int regs[26];

int base;

%}

%start list

%token DIGIT LETTER

%left '|'

%left '&'

%left '+' '-'

%left '\*' '/' '%'

%left UMINUS /\*supplies precedence for unary minus \*/

%% /\* beginning of rules section \*/

list: /\*empty \*/

|

list stat '\n'

|

list error '\n'

{

yyerrok;

}

;

stat: expr

{

printf("%d\n",$1);

}

|

LETTER '=' expr

{

regs[$1] = $3;

}

;

expr: '(' expr ')'

{

$$ = $2;

}

|

expr '\*' expr

{

$$ = $1 \* $3;

}

|

expr '/' expr

{

$$ = $1 / $3;

}

|

expr '%' expr

{

$$ = $1 % $3;

}

|

expr '+' expr

{

$$ = $1 + $3;

}

|

expr '-' expr

{

$$ = $1 - $3;

}

|

expr '&' expr

{

$$ = $1 & $3;

}

|

expr '|' expr

{

$$ = $1 | $3;

}

|

'-' expr %prec UMINUS

{

$$ = -$2;

}

|

LETTER

{

$$ = regs[$1];

}

|

number

;

number: DIGIT

{

$$ = $1;

base = ($1==0) ? 8 : 10;

} |

number DIGIT

{

$$ = base \* $1 + $2;

}

;

%%

main()

{

return(yyparse());

}

yyerror(s)

char \*s;

{

fprintf(stderr, "%s\n",s);

}

yywrap()

{

return(1);

}