

Link to github: <https://github.com/octaviaah/flcd>

lang.lxi

%option noyywrap

%option caseless

DIGIT [0-9]

WORD \"[a-zA-Z0-9]*\"

INTEGER 0|[+-]?[1-9][0-9]*

CHARACTER \"'[a-zA-Z0-9]\"'

constant {WORD}|{INTEGER}|{CHARACTER}

identifier [a-zA-Z][a-zA-Z0-9]*

%%

corner printf("Reserved word: %s\\n", yytext);

cross printf("Reserved word: %s\\n", yytext);

dribble printf("Reserved word: %s\\n", yytext);

freekick printf("Reserved word: %s\\n", yytext);

fulltime printf("Reserved word: %s\\n", yytext);

lineup printf("Reserved word: %s\\n", yytext);

score printf("Reserved word: %s\\n", yytext);

pass printf("Reserved word: %s\\n", yytext);

player printf("Reserved word: %s\\n", yytext);

referee printf("Reserved word: %s\\n", yytext);

{identifier} printf("Identifier: %s\n", yytext);

{constant} printf("Constant: %s\n", yytext);

"{" printf("Separator: %s\n", yytext);

"}" printf("Separator: %s\n", yytext);

"(" printf("Separator: %s\n", yytext);

")" printf("Separator: %s\n", yytext);

"," printf("Separator: %s\n", yytext);

"," printf("Separator: %s\n", yytext);

":" printf("Separator: %s\n", yytext);

"," printf("Separator: %s\n", yytext);

"+" printf("Operator: %s\n", yytext);

"-" printf("Operator: %s\n", yytext);

"*" printf("Operator: %s\n", yytext);

"/" printf("Operator: %s\n", yytext);

"%" printf("Operator: %s\n", yytext);

"=" printf("Operator: %s\n", yytext);

"<" printf("Operator: %s\n", yytext);

"<<" printf("Operator: %s\n", yytext);

"<=" printf("Operator: %s\n", yytext);

"==" printf("Operator: %s\n", yytext);

">=" printf("Operator: %s\n", yytext);

```

">>"      printf( "Operator: %s\n", yytext);
">"        printf( "Operator: %s\n", yytext);
"!"         printf( "Operator: %s\n", yytext);
"!="        printf( "Operator: %s\n", yytext);
"&&"        printf( "Operator: %s\n", yytext);
"||"        printf( "Operator: %s\n", yytext);
"+="        printf( "Operator: %s\n", yytext);
"-="        printf( "Operator: %s\n", yytext);
"*="        printf( "Operator: %s\n", yytext);
"/="        printf( "Operator: %s\n", yytext);

```

```

[ \t]+

```

```

[ \n]+

```

```

[+-]?0[0-9]*      printf("Illegal integer at line \n");

```

```

[0-9]+[a-zA-Z_]+[a-zA-Z0-9_]*  printf("Illegal identifier\n");

```

```

\[a-zA-Z0-9]{2,}\    printf("Character of length >=2 at line \n");

```

```

.      printf("Lexical error\n");

```

```

%%

```

```

int main( argc, argv )

```

```

int argc;

```

```

char **argv;

```

```

{

```

```
++argv, --argc;  
if ( argc > 0 )  
    yyin = fopen( argv[0], "r" );  
else  
    yyin = stdin;  
    yylex();  
}
```

p1.txt

player xa;

player ax;

pass >> ax;

cross (x <= 1) fulltime "not prime";

corner cross (x == 2) fulltime "prime";

corner cross (x % 2 == 0) fulltime "not prime";

dribble (player i = 3; i*i <=x; i += 2)

cross (x % i == 0) fulltime "not prime";

fulltime "prime";

p2.txt

player a, b, c, max;

max = 0;

pass >> a;

pass >> b;

pass >> c;

cross (a >= b && a >= c)

max = a;

corner cross (b >= a && b >= c)

max = b;

corner max = c;

score << max;