

# Lab 8 FLEX

Demo:

P1:

```
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ flex -o lang.c lang.l
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ gcc -o lang lang.c
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ ./lang lab1a/p1.txt
Separator: {
A reserved word: int
Identifier: x
Separator: ;
A reserved word: int
Identifier: y
Separator: ;
A reserved word: int
Identifier: z
Separator: ;
A reserved word: int
Identifier: max
Separator: ;
A reserved word: begin
A reserved word: read
Separator: (
Identifier: x
Separator: )
Separator: ;
A reserved word: read
Separator: (
Identifier: y
Separator: )
Separator: ;
A reserved word: read
Separator: (
Identifier: z
Separator: )
Separator: ;
Identifier: max
Operator: =
Identifier: x
Separator: ;
A reserved word: if
Separator: (
Identifier: y
Operator: >
Identifier: max
Separator: )
Separator: {
Identifier: max
Operator: =
Identifier: y
Separator: ;
Separator: }
A reserved word: if
Separator: (
Identifier: z
Operator: >
Identifier: max
Separator: )
Separator: {
Identifier: max
Operator: =
Identifier: z
Separator: ;
Separator: }
A reserved word: end
Separator: }
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$
```

```
p1.txt - Notepad
File Edit Format View Help
{
int x;
int y;
int z;
int max;

begin

read(x);
read(y);
read(z);
max=x;

if(y>max)
{
    max=y;
}
if(z>max)
{
    max=z;
}

end
}
```

Ln 10, Col 9 10

P2:

```
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc
zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ ./lang lab1a/p2.txt
Separator: {
A reserved word: int
Identifier: x
Separator: ;
A reserved word: int
Identifier: y
Separator: ;
A reserved word: begin
A reserved word: read
Separator: (
Identifier: x
Separator: )
Separator: ;
A reserved word: read
Separator: (
Identifier: y
Separator: )
Separator: ;
A reserved word: while
Separator: (
Identifier: x
Operator: !=
Identifier: y
Separator: )
Separator: {
A reserved word: if
Separator: (
Identifier: x
Operator: >
Identifier: y
Separator: )
Separator: {
Identifier: x
Operator: =
Identifier: x
Operator: -
Identifier: y
Separator: ;
Separator: }
Separator: }
A reserved word: else
Separator: {
Identifier: y
Operator: =
Identifier: y
Operator: -
Identifier: x
Separator: ;
Separator: }
Separator: }
A reserved word: write
Separator: (
Constant string: "Gcd is "
Separator: )
Separator: ;
A reserved word: write
Separator: (
Identifier: x
Separator: )
Separator: ;
A reserved word: end
Separator: }
zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$
```

```
p2.txt - Notepad
File Edit Format View Help
{
int x;
int y;

begin
read(x);
read(y);
while(x!=y)
{
    if(x>y)
    {
        x=x-y;
    }
    else
    {
        y=y-x;
    }
}
write("Gcd is ");
write(x);
end
}
```

P3:

```
zrie3047@zachman: /mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc
zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ ./lang lab1a/p3.txt
Separator: {
A reserved word: int
Separator: [
Constant number: 20
Separator: ]
Identifier: numbers
Separator: ;
A reserved word: int
Identifier: sum
Separator: ;
A reserved word: int
Identifier: index
Separator: ;
A reserved word: begin
Identifier: sum
Operator: =
Constant number: 0
Separator: ;
Identifier: index
Operator: =
Constant number: 0
Separator: ;
A reserved word: while
Separator: (
Identifier: index
Operator: <
Constant number: 20
Separator: )
Separator: {
A reserved word: read
Separator: (
Identifier: numbers
Separator: [
Identifier: index
Separator: ]
Separator: )
Identifier: index
Operator: =
Identifier: index
Constant number: +1
Separator: ;
Separator: }
Identifier: index
Operator: =
Constant number: 0
Separator: ;
A reserved word: while
Separator: (
Identifier: index
Operator: <
Constant number: 20
Separator: )
Separator: {
Identifier: sum
Operator: =
Identifier: sum
Operator: +
Identifier: numbers
Separator: [
Identifier: index
Separator: ]
Separator: ;
```

p3.txt - Notepad

File Edit Format View Help

```
{
int[20] numbers;
int sum;
int index;

begin

sum=0;
index=0;
while(index<20)
{
read(numbers[index])
index=index+1;
}

index=0;
while(index<20)
{
sum=sum+numbers[index];
index=index+1;
}

write(sum);
end
}
```

```

Identifier: sum
Operator: =
Identifier: sum
Operator: +
Identifier: numbers
Separator: [
Identifier: index
Separator: ]
Separator: ;
Identifier: index
Operator: =
Identifier: index
Constant number: +1
Separator: ;
Separator: }
A reserved word: write
Separator: (
Identifier: sum
Separator: )
Separator: ;
A reserved word: end
Separator: }
zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$

```

```

while(index<20)
{
sum=sum+numbers[index];
index=index+1;
}

write(sum);
end
}

```

## P1err:

```

zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/FLCD/lab8 lex+yacc$ ./lang lab1a/p1err.txt
Separator: {
Identifier: float
Identifier: x
Separator: ;
A reserved word: string
Identifier: str
Separator: ;
A reserved word: string
Identifier: str2
Separator: ;
A reserved word: int
Error: 324324sxcv
Separator: ;
Identifier: str
Operator: =
Error: 'lalala'
Separator: ;
Identifier: str2
Operator: =
Error: "lala
Separator: }
zrie3047@zachman:/mnt/c/Users/ztr53/Documents/GitHub/

```

p1err.txt - Notepad

File Edit Format View Help

```

{
float x;
string str;
string str2;
int 324324sxcv;

str='lalala';
str2="lala
}

```

## Flex file:

```
%option noyywrap
```

```
%{
```

```
#include <stdio.h>
```

```
%}
```

```
identifier (_|[a-zA-Z])([a-zA-Z]|[0-9])*
```

```
constnr (0|([-+]?([1-9][0-9]*)))
```

```

char [a-zA-Z ]
constchar \"\"{char}\"\"
conststring \"{char}*\"
separator \" |\"(\"|\"|\"|\"[\"|\"|\"|\"{\"|\"|\"|\";\"|\"\\n\"|\"\\t\"|\"\\r\"
operator \"+\"|\"-\"|\"*\"|\"/\"|\"<=\"|\"<\"|\">=\"|\">\"|\"!=\"|\"==\"|\"=\"|\"%\"
reserved \"char\"|\"string\"|\"int\"|\"const\"|\"if\"|\"else\"|\"read\"|\"write\"|\"while\"|\"begin\"|\"end\"

```

```

%%

```

```

[-+]?0{constnr} printf(\"Error: %s\\n\", yytext);
{constnr}{identifier} printf(\"Error: %s\\n\", yytext);
{separator} {
if(*yytext!='\\r' && *yytext!=' ' && *yytext!='\\n' && *yytext!='\\t')
printf(\"Separator: %s\\n\",yytext);
}
{operator} printf(\"Operator: %s\\n\",yytext);
{reserved} printf( \"A reserved word: %s\\n\", yytext);
{identifier} printf(\"Identifier: %s\\n\",yytext);
{constnr} printf(\"Constant number: %s\\n\",yytext);
{constchar} printf(\"Constant char: %s\\n\",yytext);
{conststring} printf(\"Constant string: %s\\n\",yytext);
(\\'{char}*\\')|(\\'{char}*)|({char}*\\')|(\\\"{char}*)|({char}*\\\") printf(\"Error: %s\\n\", yytext);

```

```

. printf(\"Error: '%s'\\n\", yytext);

```

```

%%

```

```

int main( argc, argv )
int argc;
char **argv;
{
++argv, --argc; /* skip over program name */

```

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