

LAB-1

1. Program to Recognize Relational Operators

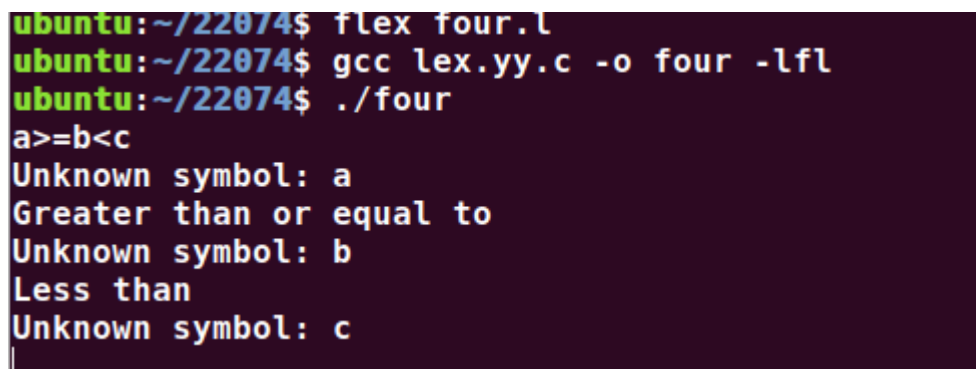
CODE:

```
%{
#include <stdio.h>
}%

%%
"=="      { printf("Equal to\n"); }
"!="      { printf("Not equal to\n"); }
">="      { printf("Greater than or equal to\n"); }
"<="      { printf("Less than or equal to\n"); }
">"       { printf("Greater than\n"); }
"<"       { printf("Less than\n"); }
[ \t\n]   { /* skip whitespace */ }
.         { printf("Unknown symbol: %s\n", yytext); }
%%

int main() {
    yylex();
    return 0;
}
```

OUTPUT:



```
ubuntu:~/22074$ flex four.l
ubuntu:~/22074$ gcc lex.yy.c -o four -lfl
ubuntu:~/22074$ ./four
a>=b<c
Unknown symbol: a
Greater than or equal to
Unknown symbol: b
Less than
Unknown symbol: c
```

2. Program to Detect Palindromes (simple word check)

CODE:

```

%{
#include <stdio.h>
#include <string.h>

int is_palindrome(char *str) {
    int len = strlen(str);
    for(int i = 0; i < len / 2; ++i) {
        if (str[i] != str[len - i - 1]) return 0;
    }
    return 1;
}
}%

%%
[a-zA-Z]+    {
                if (is_palindrome(yytext))
                    printf("Palindrome: %s\n", yytext);
            }

[ \t\n]      { /* skip whitespace */ }
.            { /* skip others */ }

%%

int main() {
    yylex();
    return 0;
}

```

OUTPUT:

```

ubuntu:~/22074$ flex five.l
ubuntu:~/22074$ gcc lex.yy.c -o five -lfl
ubuntu:~/22074$ ./five
mom
Palindrome: mom

```

3. Program to Identify Hexadecimal and Octal Numbers

CODE:

```

%{
#include <stdio.h>
%}

%%
0[xX][0-9a-fA-F]+ { printf("Hexadecimal: %s\n", yytext); }
0[0-7]+           { printf("Octal: %s\n", yytext); }
[1-9][0-9]*       { printf("Decimal: %s\n", yytext); }
[ \t\n]           { /* skip whitespace */ }
.                 { printf("Unknown token: %s\n", yytext); }
%%

int main() {
    yylex();
    return 0;
}

```

OUTPUT:

```

ubuntu:~/22074$ flex six.l
ubuntu:~/22074$ gcc lex.yy.c -o six -lfl
ubuntu:~/22074$ ./six
78A
Decimal: 78

```

4. Program to Recognize Arithmetic Operators

CODE:

```

%{
#include <stdio.h>
%}

%%
"+"      { printf("Addition operator\n"); }
"-"      { printf("Subtraction operator\n"); }
"*"      { printf("Multiplication operator\n"); }
"/"      { printf("Division operator\n"); }
"="      { printf("Assignment operator\n"); }
[0-9]+   { printf("Number: %s\n", yytext); }
[ \t\n]  { /* skip whitespace */ }
.        { printf("Unknown token: %s\n", yytext); }
%%

int main() {
    yylex();
    return 0;
}

```

OUTPUT:

```
ubuntu:~/22074$ flex seven.l
ubuntu:~/22074$ gcc lex.yy.c -o seven -lfl
ubuntu:~/22074$ ./seven
1+2-3*4/5
Number: 1
Addition operator
Number: 2
Subtraction operator
Number: 3
Multiplication operator
Number: 4
Division operator
Number: 5
```

5. Program to Recognize Valid Email Addresses

CODE:

```
%{
#include <stdio.h>
}%

%%
[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,} { printf("Valid email: %s\n",
yytext); }
.|\\n { /* skip other characters */ }
%%

int main() {
    yylex();
    return 0;
}
```

OUTPUT:

```
ubuntu:~/22074$ flex eight.l
ubuntu:~/22074$ gcc lex.yy.c -o eight -lfl
ubuntu:~/22074$ ./eight
jhguhtu@gmail.com
Valid email: jhguhtu@gmail.com
```