1.To convert an if statement multiple conditions to nested-if statements using pattern-action rules.

CODE :-

```
%option noyywrap
%{
        #include <stdio.h>
        int i, n = 0, j;
%}

%%
if\(.+\) {
        i = 0;
        while(yytext[i] != '&' && i < yyleng) {
                printf("%c", yytext[i]);
                i++;
        }
        printf(")\n");
        n++;
        while(i < yyleng) {
                i+=2;
                if(i > yyleng)
                        break;
                for(j=0; j < n; j++)
                        printf(" ");
                printf("if(");
                while(yytext[i] != '&' && i < yyleng && yytext[i] != ')') {
                        printf("%c", yytext[i]);
                        i++;
                }
                n++;
                printf(")\n");
        }
}
.*; {
        for(j=1; j < n; j++)
               printf(" ");
        printf("%s", yytext);
}
.|\n
%%

int main() {
        FILE* fd = fopen("input.txt", "r");
        yyin = fd;
        yylex();
        return 0;
}
```

OUTPUT :-

```
C:\WINDOWS\system32\cmd.exe

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>lex p1.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>gcc lex.yy.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>a.exe
if(x == 5 )
 if( y == 10 )
  if( z == 15)
        statement;
```

```
input.txt - Notepad
File  Edit  Format  View  Help
if(x == 5 && y == 10 && z == 15)
        statement;
```

2.Convert nested if-else to nested do - while syntax using LEX.

CODE :-

```
%option noyywrap
%{
        #include<string.h>
        int n = 0, i, j, m;
        char stack[10][1024];
%}
%%
if\(.+\) {
         printf("while(");
         j = 0;
         for(i = 3; i < yyleng-1; i++) {
                 printf("%c", yytext[i]);
                 if(yytext[i] == '=' && yytext[i+1] == '=') {
                         stack[n][j++] = '!';
                 }
                 else {
                         stack[n][j++] = yytext[i];
                 }
         }
         printf(") {");
         n++;
}
```

```
.*; {
        for(i=0; i < yyleng && yytext[i] == ' '; i++);
        m = i / 4;

        printf("%s\n", yytext);
        for(i=0; i<m; i++) {
              printf(" ");
        }
        printf("break;\n");
        for(i = 0; i < m-1; i++) {
              printf(" ");
        }
        printf("}");
        n--;
}
.*else {
        for(i=0; i < yyleng && yytext[i] == ' '; i++);
        m = i / 4;
        for(i = m; i < n; i++) {
                for(j = 0; j < n - i; j++) {
                      printf(" ");
                }
                printf("break;\n");
                for(j = 1; j < n - i; j++) {
                      printf(" ");
                }
                printf("}\n");
        }
        n = m;
        for(i=0; i<m; i++) {
              printf(" ");
        }
        printf("while(%s) {", stack[n]);
        n++;
}
%%
int main() {
        FILE *fd = fopen("input2.txt", "r");
        yyin = fd;
        yylex();
        printf("\n");
        for(i = 0; i < n; i++) {
                for(j = 0; j < n - i; j++) {
                      printf(" ");
                }
                printf("break;\n");
                for(j = 1; j < n - i; j++) {
                      printf(" ");
                }
                printf("}\n");
        }
        return 0;
}
```

OUTPUT :-

```
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>lex p2.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>gcc lex.yy.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>a.exe
while( x == 1 ) {
        while( y == 2 ) {
                statement 1;
break;
}
 break;
}
while( x != 1 ) {
                statement 2;
break;
}
while( x != 1 ) {
        while( z == 3 ) {
                statement 3;
break;
}
 break;
}
while( x != 1 ) {
                statement 4;
break;
}
```

3.Convert a nested if –else if consisting of multiple expressions into a single if-else if

CODE :-

```
%option noyywrap
%{
        #include <stdio.h>
        int i, n = 0, j;
%}

%%
if\(.+\) {
        if(n == 0) {
                for(i = 0; i < yyleng-1; i++) {
                        printf("%c", yytext[i]);
                }
                n++;
        }
        else {
                printf(" && ");
                for(i = 3; i < yyleng-1; i++) {
                        printf("%c", yytext[i]);
                }
                n++;
       }
}
else.if\(.+\) {
        n = 1;
        printf("\n");
        for(i = 0; i < yyleng-1; i++) {
              printf("%c", yytext[i]);
        }
}
.*; {
        printf(")\n");
        i = 0;
        while(yytext[i] == ' ' && i < yyleng) {
              i++;
        }
        printf(" ");
        while(i < yyleng) {
                printf("%c", yytext[i]);
                i++;
        }
}
.|\n
%%

int main() {
        FILE* fd = fopen("inp.txt", "r");
        yyin = fd;
        yylex();
        return 0;
}
```

OUTPUT :-

```
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>lex p3.l

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>gcc lex.yy.c

D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>a.exe
if(a == 10 || c == 30 && b == 20)
                statement 1;
else if(d == 40 && e == 50)
                statement 2;
D:\STUDIES\SEM 5\CD\LAB\CODE\LAB 7>
```