12-10-2021 COMPILER DESIGN SACHIN RAGHUL T

LAB - 06 2019103573

OBSERVATION

1. Replace the switch with if else statement

```
%option noyywrap
%{
    #include <stdio.h>
    #include <string.h>
    int n = 0, i ;
    char ex[20], val[20] ;
%}
switch
\(.+\) {
        for(i = 0; i < yyleng - 2; i++)
            ex[i] = yytext[i+1];
        ex[i] = '\0';
}
case .* {

        for (i = 0 ; i << yyleng-5; i++)
            val[i] = yytext[i+4];
        val[i] = '\0';
        if(n==0){
            printf (" if (%s==%s)", ex,val);
            n++;
        }
        else {
            printf("  else if (%s= =%s)", ex, val);
            n++;
        }
}
break;
default: {
        printf (" (else");
}
\{|\}
%%
```

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

2) convert a function to a macro

```
% option noyywrap
%{
  #include <stdio.h>
  #include <string.h>
  char macro[100];
  char result[100] = "#define ";
  int i, l=0;
%}
%%
int, [a-z A-Z] + [\(,\)]{
   for(i=4; i<yyleng ; i++) {
       macro [l++] = yytext[i];
   }
}
return .*; {
   strcat(macro, "(");
   l += 2;
   for(i=7 ; i<yyleng -1 ;i++) {
       macro[l++] = yytext[i];
   }
   strcat(macro , ")");
}
% \n
%%
```

```
int main()
{
    FILE *fd = fopen("function.txt", "r");
    yyin = fd;
    yylex();
    strcat(result, macro);
    printf("%s\n", result);
    return 0;
}
```

3. convert infix to postfix.

```
%option noyywrap
%{
  #include <stdio.h>
  int top = 0;
  char stack[100];
  int prec(char c) {
    if(c = 'r') {
      return 3;
    }
    else if(c == '/' || c == '*') {
      return 2;
    }
    else if(c == '+' || c == '-') {
      return 1;
    }
    else
      return -1;
  }
%}
```

```
%%
[a-z] { printf (" %c", yytext[0]); }
\^|\+|-|\*|\/ {
    while(top >0 && prec (yytext (yytext [0] )<= prec(stack[top-1])
        printf (" %c", stack[top-1]);
        top--;
    }
    stack [top ++] = yytext[0];
}
\( { stack [top ++] = yytext [0] ; }
\) {
    while (top >0 && stack[top-1] != '(' ){
        printf (" %c", stack[top-1] );
        top--;
    }
}
\n {
   while (top>0) {
     if (stack[top - 1] == '(' ){
        top--;
        continue;
     }
     printf(" %c", stack[top-1]);
     top--;
   }
   top = 0;
   printf ("\n\n");
}
%%
int main()
{
    yylex();
    return 0;
}
```