Khulna University of Engineering & Technology Department of Computer Science & Engineering

Course No: CSE 3212

Course Name: Compiler Design Laboratory

Submission By:

Faisal Ahmed

Roll: 1607048

Group: A2

Compiler

A compiler is a computer program that translates computer code written in one programming language into another language. The name compiler is primarily used for programs that translate source code from a high-level programming language to a lower level language to create an executable program

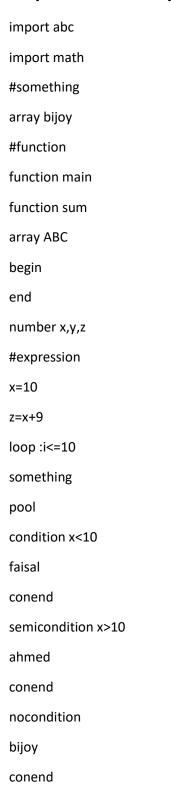
Lex

- 1.Lex is a program that generates lexical analyzer. It is used with YACC parser generator.
- 2.The lexical analyzer is a program that transforms an input stream into a sequence of tokens.
- 3.It reads the input stream and produces the source code as output through implementing the lexical analyzer in the C program.

Instruction in cmd

- 1.flex Faisal.l
- 2. gcc lex.yy.c -o app
- 3.app

Input file :input.txt



Output file : ouput.txt

header found :: abc
header found :: math
Comment found : #something
array assign : array bijoy
Comment found : #function
function called
main
function called
sum
array assign : array ABC
keyword Found : begin
keyword Found : end
varriable declared
x
У
y z
•
Z
z Comment found : #expression
z Comment found : #expression value assign
z Comment found : #expression value assign Mathematical expression
z Comment found: #expression value assign Mathematical expression loop found
z Comment found : #expression value assign Mathematical expression loop found Inside Loop
z Comment found : #expression value assign Mathematical expression loop found Inside Loop Loop ended
z Comment found : #expression value assign Mathematical expression loop found Inside Loop Loop ended if condition
z Comment found : #expression value assign Mathematical expression loop found Inside Loop Loop ended if condition Inside condition
z Comment found : #expression value assign Mathematical expression loop found Inside Loop Loop ended if condition Inside condition condition ended

```
else condition
Inside condition
condition ended
Program ended
------
Number of header file: 2
Number of Loop: 1
Number of condition: 3
Number of function: 2
Number of array: 2
Number of expression: 2
```

Code Faisal.l

Number of declared number: 1

Number of Comment: 3

```
#include<string.h>
int loopst=0;
int condition=0;
int headercount=0;
int loopcount=0;
int conditioncount=0;
int functioncount=0;
int arraycount=0;
int expressioncount=0;
int numbercount=0;
int commentcount=0;
```

```
varriable [a-z|A-Z]+[a-z|A-Z|0-9]*
 \text{Operator "="|"+"|"-"|"/"|"*"|"&"|"|"|"%"} \\
Releational\_Operator ">"|"<"|"<="|">="|"&&"|"||"|"=="|"!="|
digit [0-9]
keyword "begin" | "end"
%%
"#".* {printf("Comment found : %s\n",yytext);
                 commentcount++;
}
"import ".* {
  printf("header found :: ");
  int i;
  headercount++;
  for(i=7;i < strlen(yytext);i++){}
    printf("%c",yytext[i]);
  printf("\n");
}
\label{linear} $$ {\rm varriable}{Operator}{digit}^* \{ 
        printf("value assign\n");
        expressioncount++;
}
{varriable}{Operator}{varriable}{Operator}{digit}* {
        printf("Mathematical expression\n");
        expressioncount++;
}
{keyword}.* {
```

```
printf("keyword Found : %s",yytext);
}
"array ".* {
        printf("array assign : %s\n",yytext);
        arraycount++;
}
"function ".* {
  printf("function called\n");
  int i;
  functioncount++;
  for(i=9;i<strlen(yytext);i++){</pre>
     printf("%c",yytext[i]);
  }
  printf("\n");
}
"number "({varriable},)*{varriable} {
  printf("varriable declared\n");
  int i;
  numbercount++;
  for(i=7;i < strlen(yytext);i++){}
    if(yytext[i]==','){
       printf("\n");
    }
     else{
       printf("%c",yytext[i]);
```

```
}
           }
}
"loop:"{varriable}{Releational\_Operator}({digit})^* | {varriable}^*) \{
           printf("loop found\n");
           loopst++;
           loopcount++;
}
"pool" {
           loopst--;
           if(loopst<1){
                       printf("Loop ended\n");
           }
}
"conend" {
           condition--;
           printf("condition ended\n");
}
"condition "\{varriable\}[" "]*\{Releational\_Operator\}[" "]*(\{digit\}* | \{varriable\}*) \{ for each of the property of the propert
           printf("if condition\n");
           condition++;
           conditioncount++;
}
"semicondition "{varriable}[" "]*{Releational_Operator}[" "]*({digit}*|{varriable}*) {
           printf("elseif condition\n");
```

```
condition++;
  conditioncount++;
}
"nocondition" {
  printf("else condition\n");
  condition++;
  conditioncount++;
}
.* {
  if(loopst==0 && condition==0){
    printf("%s",yytext);
  }
  if(loopst){
    printf("Inside Loop");
  }
  if(condition){
    printf("Inside condition");
 }
}
%%
int yywrap()
{
  return 1;
}
int main(){
```

```
freopen("input.txt","r",stdin);
freopen("output.txt","w",stdout);
yylex();
printf("Program ended\n");
printf("-----\n");
printf("Number of header file : %d\n",headercount);
printf("Number of Loop : %d\n",loopcount);
printf("Number of condition : %d\n",conditioncount);
printf("Number of function : %d\n",functioncount);
printf("Number of array : %d\n",arraycount);
printf("Number of expression : %d\n",expressioncount);
printf("Number of declared number : %d\n",numbercount);
printf("Number of Comment : %d\n",commentcount);
return 0;
}
```