PROGRAM – 1

```
%{
int vowel_count = 0;
int consonant_count = 0;
%}
%%
[aAeEiIoOuU]      { vowel_count++; }
[a-zA-Z]          { consonant_count++; }
\n                { return 0; }
.                 ;
%%
int main()
{
   yylex();
   printf("Number of vowels: %d\n", vowel_count);
   printf("Number of consonants: %d\n", consonant_count);
   return 0;
}
int yywrap()
{
   return 1;
}
```

PROGRAM – 2

```
%{
#include<stdio.h>
int lines=0, words=0, c_letters=0, num=0, spl_chr=0, total=0, blank_spaces=0;
%}
%%
\n          { lines++; words++; }
[\t]        { words++; }
[ ]         { blank_spaces++; words++; }
[a-zA-Z]    { c_letters++; }
[0-9]       { num++; }
.           { spl_chr++; }
%%
int main()
{
   FILE *file;
   file = fopen("sample2.txt", "r");
   if (!file){
   printf("Error: could not open file 'sample.txt'\n");
   return 1;
   }
   yyin = file;
   yylex();
   fclose(file);
   total = c_letters + num + spl_chr;
   printf("\nFile Contents...\n");
   printf("\n\t%d lines", lines);
   printf("\n\t%d words", words);
   printf("\n\t%d digits", num);
   printf("\n\t%d alphabetic characters", c_letters);
```

```
    printf("\n\t%d special characters", spl_chr);
    printf("\n\t%d blank spaces", blank_spaces);
    printf("\n\n\tTotal %d characters\n", total);
    return 0;
}
int yywrap()
{
    return 1;
}
```

#gedit sample2.txt

```
Hi  Hello.
123
```

PROGRAM – 3

```
%{
 #include<stdio.h>
 int posint=0, negint=0,posfraction=0, negfraction=0;
%}
%%
[-][0-9]+               {negint++;}
[+]?[0-9]+              {posint++;}
[+]?[0-9]*\.[0-9]+      {posfraction++;}
[-][0-9]*\.[0-9]+       {negfraction++;}
%%
int main()
{
 FILE *file;
    file = fopen("sample3.txt", "r");
    if (!file) {
        printf("Error: could not open file 'sample3.txt'\n");
        return 1;
    }
    yyin = file;
    yylex();
    fclose(file);
printf("\nNumber of +ve integers=%d\n Number of –ve integers=%d\n Number of +ve
fractions=%d\n Number of –ve fractions=%d\n", posint, negint,posfraction, negfraction);
}
int yywrap()
{
    return 1;
}
```

#gedit sample3.txt

83

-37

0.5

-0.8 -1.0

PROGRAM – 4

```
%{
#include <stdio.h>
int com = 0;
%}
%%
"//".*    { com++; }
%%
int main()
{
   FILE *file;
   file = fopen("sample4.txt", "r");
   if (!file) {
      printf("Error: could not open file 'sample4.txt'\n");
      return 1;
   }
   yyin = file;
   yylex();
   fclose(file);
   printf("Number of comment lines = %d\n", com);
   return 0;
}
int yywrap()
{
   return 1;
}
```

#gedit sample4.txt

```
void main()
{
        int a;  // integer variable
        float b;  // float variable
        // multiline
}
```

PROGRAM – 5

```
%{
#include <stdio.h>
int printf_count = 0;
int scanf_count = 0;
%}
printf      printf
scanf       scanf
%%
{printf}     {
              printf_count++;
              printf("std::cout");
               }
{scanf}      {
              scanf_count++;
              printf("std::cin");
               }
.|\n         {
              printf("%s", yytext);
             }
%%
int main()
{
   FILE *file;
   file = fopen("sample5.txt", "r");
   if (!file) {
      printf("Error: could not open file 'sample5.txt'\n");
      return 1;
   }
   yyin = file;
```

```
    yylex();
    fclose(file);
    printf("\nNumber of printf statements: %d\n", printf_count);
    printf("Number of scanf statements: %d\n", scanf_count);
    return 0;
}
int yywrap()
{
    return 1;
}
```

#gedit sample5.txt

```
#include <stdio.h>
int main()
{
    int a, b;
    printf("Enter two numbers: ");
    scanf("%d %d", &a, &b);
    printf("You entered: %d and %d\n", a, b);
    if (a > b) {
        printf("a is greater than b\n");
    } else {
        printf("b is greater than or equal to a\n");
    }
    return 0;
}
```

PROGRAM – 6

```
%{
#include<stdio.h>
#include<string.h>
int i=0, o=0, flag=0, k;
char id[10][10], op[10][10];
%}
%%
[0-9]+      { flag++; strcpy(id[i], yytext); i++; }
[+*-/]      { flag--; strcpy(op[o], yytext); o++; }
.|\n        { return 0; }
%%
int main()
{
    printf("Enter the expression: \n");
    yylex();
    if(flag!=1){
        printf("\n Invalid expression\n");
    }
    Else
    {
        printf("\n Valid expression\n");
        printf("\n Operators are:\n");
        for (k=0;k<o;k++)
        {
            printf("%s\t", op[k]);
        }
        printf("\n Identifiers are:\n");
        for (k=0;k<i;k++)
        {
```

```
                printf("%s\t", id[k]);
            }
        }
}
int yywrap()
{
    return 1;
}
```

PROGRAM – 7

```
%{
#include<stdio.h>
int flag=0;
%}
%%
and |
or  |
but |
like |
then {flag=1;}
.    ;
\n   {return 1;}
%%
int main()
{
   printf("Enter the sentence:\n");
   yylex();
   if (flag==0){
      printf("\nSimple sentence\n");
   }
   else {
      printf("\nCompound sentence\n");
   }
}
int yywrap()
{
   return 1;
}
```

PROGRAM – 8

```
%{
#include <stdio.h>
int identifier_count = 0;
%}
%%
[a-zA-Z][a-zA-Z0-9]*    { identifier_count++; }
.
\n
%%
int main() {
   FILE *file;
   file = fopen("sample8.txt", "r");
   if (!file) {
      printf("Error: could not open file 'sample8.txt'\n");
      return 1;
   }
   yyin = file;
   yylex();
   fclose(file);
   printf("Number of identifiers: %d\n", identifier_count);
   return 0;
}
int yywrap() {
   return 1;
}
```

#gedit sample8.txt

```
a=1
b=2
c=3
```

PROGRAM – 9

```
%{
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define MAX_SYMBOLS 100

typedef struct {
    char name[100];
    int line_no;
} symbol;
symbol symboltable[MAX_SYMBOLS];
int symbolcount = 0;
void addsymbol(char *name, int line_no);
void printsymboltable();
extern int yylineno;
%}
%%
[a-zA-Z_][a-zA-Z0-9_]* {addsymbol(yytext, yylineno);}
\n                     {yylineno++;}
[ \t\n]                ;
.                      ;
%%
void addsymbol(char *name, int line_no)
{
    for (int i = 0; i < symbolcount; ++i)
    {
        if (strcmp(symboltable[i].name, name) == 0)  {
            return;
        }
```

```
    }
    if (symbolcount < MAX_SYMBOLS)
    {
        strcpy(symboltable[symbolcount].name, name);
        symboltable[symbolcount].line_no = line_no;
        ++symbolcount;
    }
    else
    {
        fprintf(stderr, "Symbol table overflow\n");
    }
}
void printsymboltable()
{
    printf("Symbol Table:\n");
    printf("Name\t\tLine Number\n");
    printf("----\t\t-----------\n");
    for (int i = 0; i < symbolcount; ++i)
    {
        printf("%s\t\t%d\n", symboltable[i].name, symboltable[i].line_no);
    }
}
int main()
{
    FILE *file;
    file = fopen("sample9.txt", "r");
    if (!file) {
        printf("Error: could not open file 'sample9.txt'\n");
        return 1;
    }
```

```
    yyin = file;
    yylineno=1;
    yylex();
    fclose(file);
    printsymboltable();
    return 0;
}
int yywrap()
{
    return 1;
}
```

#gedit sample9.txt

```
a
b
d
c
e
f
```