LEX PROGRAMS FOR THE CLASS TEST DATED 30th SEPTEMBER SCHEDULED AT 4-5 pm

1. Write a lex program to find the number of vowels and consonants.

```
%{
/* to find vowels and consonants*/
int vowels = 0;
int consonants = 0;
%}
%%
[ \t\n]+
[aeiouAEIOU] vowels++;
[bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ] consonants++;.
%%
main()
{
yylex();
printf(" The number of vowels = %d\n", vowels);
printf(" number of consonants = %d \n", consonants);
return(0);
}
```

2. Write a lex program to find the number of positive integer, negative integer, positive floating positive number and negative floating point number.

```
%{
int posnum = 0;
int negnum = 0;
int posflo = 0;
int negflo = 0;
%}
%%
[\n\t ];
([0-9]+) {posnum++;}
-?([0-9]+) {negnum++; }
([0-9]*\.[0-9]+) { posflo++; }
-?([0-9]*\.[0-9]+) { negflo++; }
. ECHO;
%%
main()
{
yylex();
```

```
printf("Number of positive numbers = %d\n", posnum);
printf("number of negative numbers = %d\n", negnum);
printf("number of floting positive number = %d\n", posflo);
printf("number of floating negative number = %d\n", negflo);
}
```

3. Write a lex program to replace scanf with READ and printf with WRITE statement also find the number of scanf and printf.

```
%{
int pc=0,sc=0;
%}
%%
printf fprintf(yyout,"WRITE");pc++;
scanf fprintf(yyout,"READ");sc++;
. ECHO;
%%
main(int argc,char* argv[])
{
if(argc!=3)
{
printf("\nUsage: %s <src> <dest>\n",argv[0]);
return;
}
yyin=fopen(argv[1],"r");
yyout=fopen(argv[2],"w");
yylex();
printf("\nno. of printfs:%d\nno. of scanfs:%d\n",pc,sc);
}
```

4. Program to count the number of characters, words, spaces and lines in a given input file.

```
%{
#include<stdio.h>
int wc=0,cc=0,lc=0,bc=0;
%}
%%
[a-zA-Z]* {wc++; cc=cc+yyleng ;}
"\n" {lc++ ;}
[ ] {bc++ ;}
. {cc++ ;}
%%
int main()
```

```
{
FILE *fp;
char file[50];
printf("ENTER THE FILENAME:\n");
scanf("%s",file);
fp=fopen(file,"r");
if(!fp)
{
fprintf(stderr,"FILE DOES NOT EXIST\n");
exit(1);
}
yyin=fp;
yylex();
printf (" NO OF CHARACTERS=%d\n NO OF WORDS=%d\n NO OF LINES=%d\n NO OF
BLANKS=%d\n",cc,wc,lc,bc);
return 0;
}
```

5. Program to recognize and count the number of identifiers in a given input file.

```
%{
#include<stdio.h>
int count=0;
%}
op [+-*/]
letter [a-zA-Z]
digit [0-9]
id {letter}+({letter}|{digit})*
notid {digit}+{id}
%%
[\t\n]+ ;
("int") |
("float") |
("char") |
("case") |
("default") |
("if") |
(“else”) |
(“then”) |
(“while”) |
("for") |
```

```
("printf") |
("scanf") {printf("%s IS A KEYWORD\n",yytext);}
{id} {printf("%s IS AN IDENTIFIER\n",yytext);count++;}
{notid} {printf("%s IS NOT AN IDENTIFIER\n",yytext);}
%%
int main()
{
FILE *fp;
char file[10];
printf("ENTER THE FILENAME:\n");
scanf("%s",file);
fp=fopen(file,"r");
yyin=fp;
yylex();
printf("TOTAL IDENTIFIERS ARE=%d\n",count);
return 0;
}
```

6.lex program to find thye longest word

```
%{
#include <strings.h>

int longest = 0;
char longword[60];
%}

%%
[a-zA-Z]+           { if (yyleng > longest) {
                      longest = yyleng;
                      strcpy (longword, yytext);
                      }
                    }
.                   |
\n                  ;
%%
int main (void) {
  yylex ();
  printf ("The longest word was \"%s\", which was %d characters long.\n",
             longword, longest);
  return 0;
  }
```