6. a) Write a LEX program to eliminate *comment lines* in a *C* program and copy the resulting program into a separate file.

6a.l

```
%{
#include<stdio.h>
int ccount=0;
%}
%%
"/*"[^*/]*"*/"          {ccount++;}
"//".*                  {ccount++;}
%%
int main(int argc,char **argv)
{
        FILE *f1,*f2;
        if(argc>1)
        {
                f1=fopen(argv[1],"r");
                if(!f1)
                {
                        printf("Error in opening input file\n");
                        exit(1);
                }
                yyin=f1;
                f2=fopen(argv[2],"w");
                if(!f2)
                {
                        printf("Error in opening output file\n");
                        exit(1);
                }
                yyout=f2;
                yylex();
                printf("Number of comment lines:%d\n",ccount);
        }
}
```

Input

1.c

```
/* program to add two numbers */
#include<stdio.h>
int main()
{
        int a=10,b=20; //declaring and initializing
        int sum;
        sum=a+b; //computing sum
        printf(“sum is %d”,sum); //display sum
}
```

Execution Steps:

```
lex 6a.l
cc lex.yy.c  –ll
. /a.out   1.c    2.c
```

b) Write YACC program to recognize valid *identifier, operators and keywords* in the given text (*C program*) file.

6b.l

```
%{
#include<stdio.h>
#include"y.tab.h"
%}
%%
[\t]                    ;
[+|-|*|/|=|<|>]      {printf("Operator:%s\n",yytext);return OP;}
[0-9]+               {printf("Numbers:%s\n",yytext);return NUM;}
int|char|bool|float|void|for|do|while|if|else|return    {printf("Keyword:%s\n",yytext);return KEY;}
[a-zA-Z][a-zA-Z0-9]*        {printf("Identifier:%s\n",yytext);return ID;}
\"[^"]*\"                         ;
.                                 ;
%%
```

6b.y

```
%{
#include<stdio.h>
#include<stdlib.h>
extern FILE *yyin;
int id=0,dig=0,key=0,op=0;
%}
%token NUM  ID KEY OP
%%
input:NUM  input      {dig++;}
|ID input             {id++;}
|KEY input            {key++;}
|OP input             {op++;}
|NUM                  {dig++;}
```

```
|ID           {id++;}
|KEY          {key++;}
|OP           {op++;}
;
%%
main()
{
        FILE *myfile=fopen("input.c","r");
        if(!myfile)
        {
                printf("Error in opening input.c!");
                return-1;
        }
        yyin=myfile;
        yyparse();
        printf("numbers= %d\nKeywords= %d\nIdentifiers= %d\noperators= %d\n",dig,key,id,
        op);
}
void yyerror()
{
        printf("Parse error!");
        exit(-1);
}
```

<u>Input</u>

input.c

```
#include<stdio.h>
int main()
{
        int a,b;
        printf(“enter a value”);
        scanf(“%d”,&a);
        printf(“enter b value”);
        scanf(“%d”,&b);
        if(a>b)
                printf(“a is greater”);
        else
                printf(“b is greater”);
        return 0;
}
```

Execution Steps:

```
lex 6b.l
yacc –d 6b.y
cc lex.yy.c  y.tab.c –ll
. /a.out
```