YVONNE MARKHIZE STANLEY DENNIS FLO

1) **Write a LEX specification files to:**
   a) **Count the number of words in a file and their total size**

Solution

```
%{
        #include <stdio.h>
        int lines=0, words=0, c_letters=0, num=0, spl_chr=0, total=0;
%}

%%
\n {lines++; words++;}
[\t ' '] words++;
[a-zA-Z] c_letters++;
[0-9] num++;
. spl_chr++;
%%

int main(){
        yyin=fopen("sample.txt","r");
        yylex();
        total = c_letters+num+spl_chr;
        printf("File Contents...\n");
        printf("\n\t%d Lines",lines);
        printf("\n\t%d Words",words);
        printf("\n\t%d Text Characters",c_letters);
        printf("\n\t%d Digits",num);
        printf("\n\t%d Special Characters",spl_chr);
        printf("\n\t%d Total Characters",total);
        return 0;
}
int yywrap(){
        return 1;
}
```

   b) **Counts the number of different words in an input**

Solution

   c) **Accepts the English language words (without bothering for the meaning) and replaces each occurrence of the string “abc” in it to “ABC”.**

Solution

```
%{
#include<stdio.h>
#include<string.h>
int i;
%}
```

```
%%

[a-zA-Z]* {
                                        for(i=0;i<=yyleng;i++)
                                        {
                                                if((yytext[i]=='a')&&(yytext[i+1]=='b')&&(yytext[i+2]=='c'))
                                                {
                                                        yytext[i]='A';
                                                        yytext[i+1]='B';
                                                        yytext[i+2]='C';
                                                }
                                        }
                                        printf("%s",yytext);
                                }

.*                      {ECHO;}
\n                      {printf("%s",yytext);}

%%

int main()
{
        yylex();
        return 0;
}

int yywrap()
{
        return 1;
}
```

2) The following is a listing of a set of verbs:

is am are were *do does did will* *has have had go*

was be being been *would should can could*

Write a simple LEX specification to recognize these verbs

Solution

```
%{
    #include <stdio.h>
%}

%%
```

YVONNE MARKHIZE STANLEY DENNIS FLO

```
[\t ' ']+                /* ignore whitespace */ ;

is |
am |
are |
were |
was |
be |
being |
been |
do |
does |
did  |
will |
would |
should |
can  |
could |
has  |
have |
had |
go       { printf("%s: is a verb\n", yytext); }
[a-zA-Z]+ { printf("%s: is not a verb\n", yytext); }

.|\n     { ECHO; /* normal default anyway */ }
%%

int main()
{
     yylex() ;
     return 0;
}

int yywrap(){
       return 1;
}
```