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

*Solution*

%{

#include <stdio.h>

int words=0, c\_letters=0, total=0;

%}

%%

\n {words++;}

[\t ' '] words++;

[a-zA-Z] c\_letters++;

%%

int main(){

yyin=fopen("sample.txt","r");

yylex();

total = c\_letters+words;

printf("File Contents...\n");

printf("\n\t%d Words",words);

printf("\n\t%d Text Characters",c\_letters);

printf("\n\t%d Total Characters\n",total);

return 0;

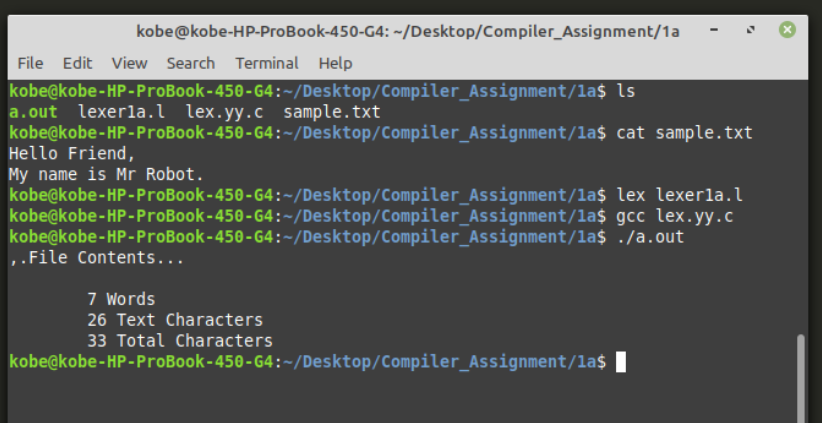
}

int yywrap(){

return 1;

}

*Result*

**

* 1. **Counts the number of different words in an input**

*Solution*

* 1. **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;

}

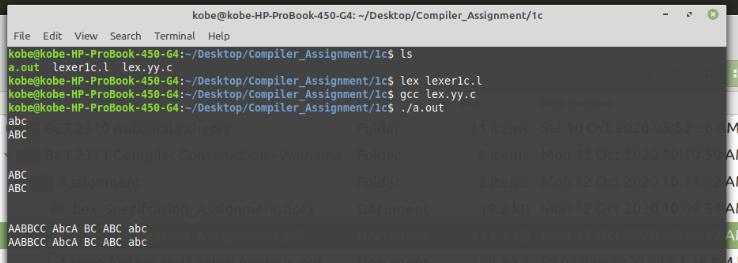
int yywrap()

{

return 1;

}

*Result*

**

1. **The following is a listing of a set of verbs:**

*is am are were*

*was be being been*

*do does did will*

*would should can could*

*has have had go*

**Write a simple LEX specification to recognize these verbs**

*Solution*

%{

#include <stdio.h>

%}

%%

[\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;

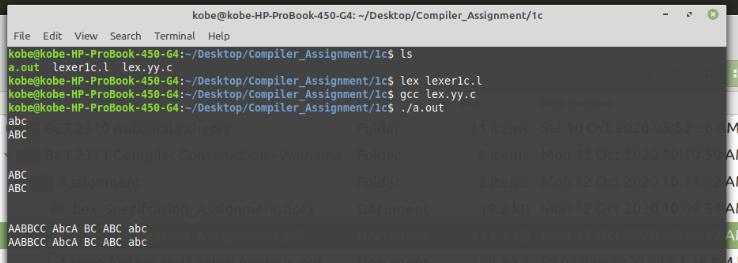
}

int yywrap(){

return 1;

}

*Result*

**