**Code for compilation:**

Compilation of lex file

lex (filename.l)

Compilation of C file generated by previous lex file compilation

gcc lex.yy.c -lfl

Execute the output file

./a.out

**Code:**

**First.l**

%{

%}

%%

\n {

printf("Hello Good morning");

}

%%

void main(){

yylex();

}

**Second.l**

%{

char username[30];

int continueprogram;

%}

%%

[\n] {

printf("Hello %s Good morning", username);

return 1;

}

%%

void main(){

do

{

printf("Enter your first name: ");

scanf("%s", username);

yylex();

printf("\nEnter 1 to redo or 0 to exit the program: \n");

scanf("%d", &continueprogram);

}while(continueprogram == 1);

}

**Third.l**

%{

int linecount=0,charcount=0;

%}

%%

. {charcount++;}

\n {linecount++; charcount++;}

%%

void main(){

yylex();

printf("# of lines = %d, # of chars = %d", linecount,charcount);

}

**Fourth.l**

%{

void display(int, char \*);

int flag;

%}

%%

[a|e|i|o|u] { flag =1; display(flag,yytext); }

. { flag =0; display(flag,yytext); }

%%

void main()

{

printf("\nEnter the word:"); yylex();

}

void display(int flag,char \*t)

{

if(flag==1)

{

printf("\nThe given character %s is vowel \n",t);

}

else

{

printf("\nThe given character %s is not vowel \n",t);

}

}

**Fifth.l**

%{

int COMMENT=0;

%}

identifier [a-zA-Z][a-zA-Z0-9]\*

%%

#.\* {printf("\n%s is a preprocessor directive",yytext);}

int |

float |

char |

double |

while |

for |

struct |

typedef |

do |

if |

break |

continue |

void |

switch |

return |

else |

goto {printf("%s is a keyword",yytext);}

"/\*" {COMMENT=1;}{printf("\n%s is a comment",yytext);}

{identifier}\( {if(!COMMENT)printf("\n%s is a function",yytext);}

{identifier}(\[[0-9]\*\])? {if(!COMMENT) printf("\n%s identifier",yytext);}

\".\*\" {if(!COMMENT)printf("\n%s is a string",yytext);}

[0-9]+ {if(!COMMENT) printf("\n%s is a number ",yytext);}

\)(\:)? {if(!COMMENT)printf("\n");ECHO;printf("\n");}

\( ECHO;

= |

\<= |

\>= |

\< |

== |

\> {if(!COMMENT) printf("\n%s is a an operator",yytext);}

%%

int main(int argc, char \*\*argv)

{

FILE \*file;

file=fopen("code.c","r");

if(!file)

{

printf("could not open the file");

exit(0);

}

yyin=file;

yylex();

printf("\n");

return(0);

}

int yywrap()

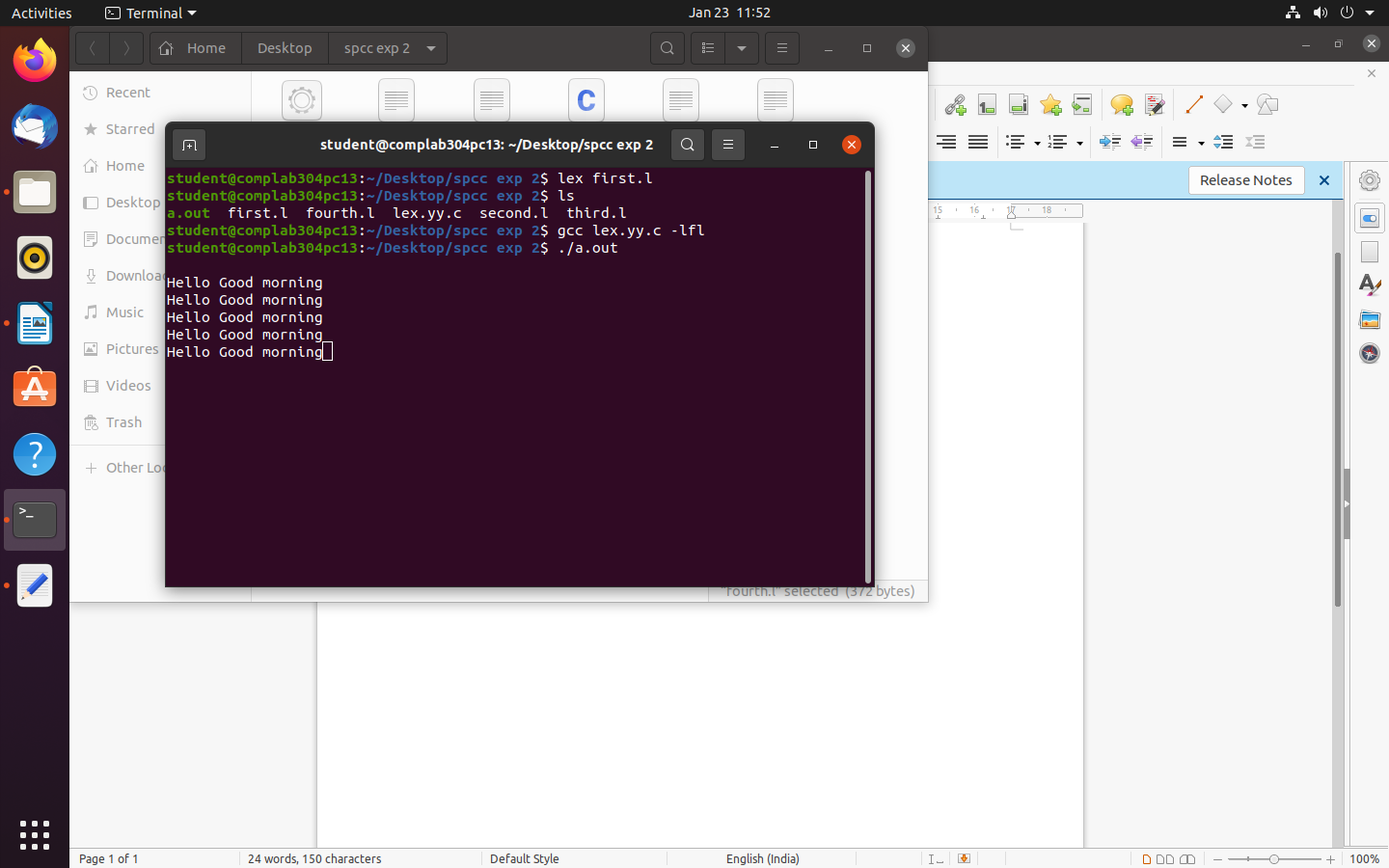
{

return(1);

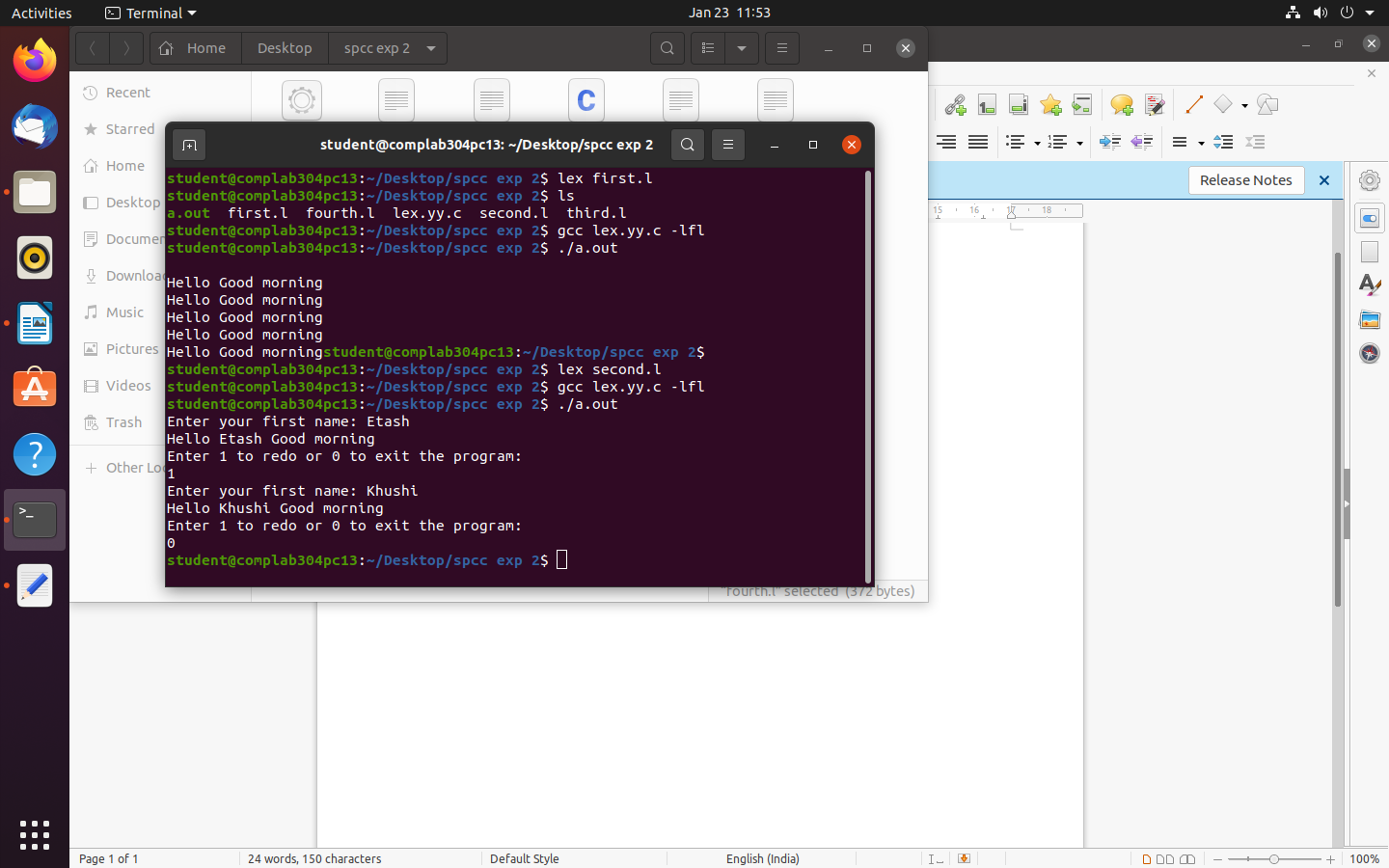
}

**Output:**

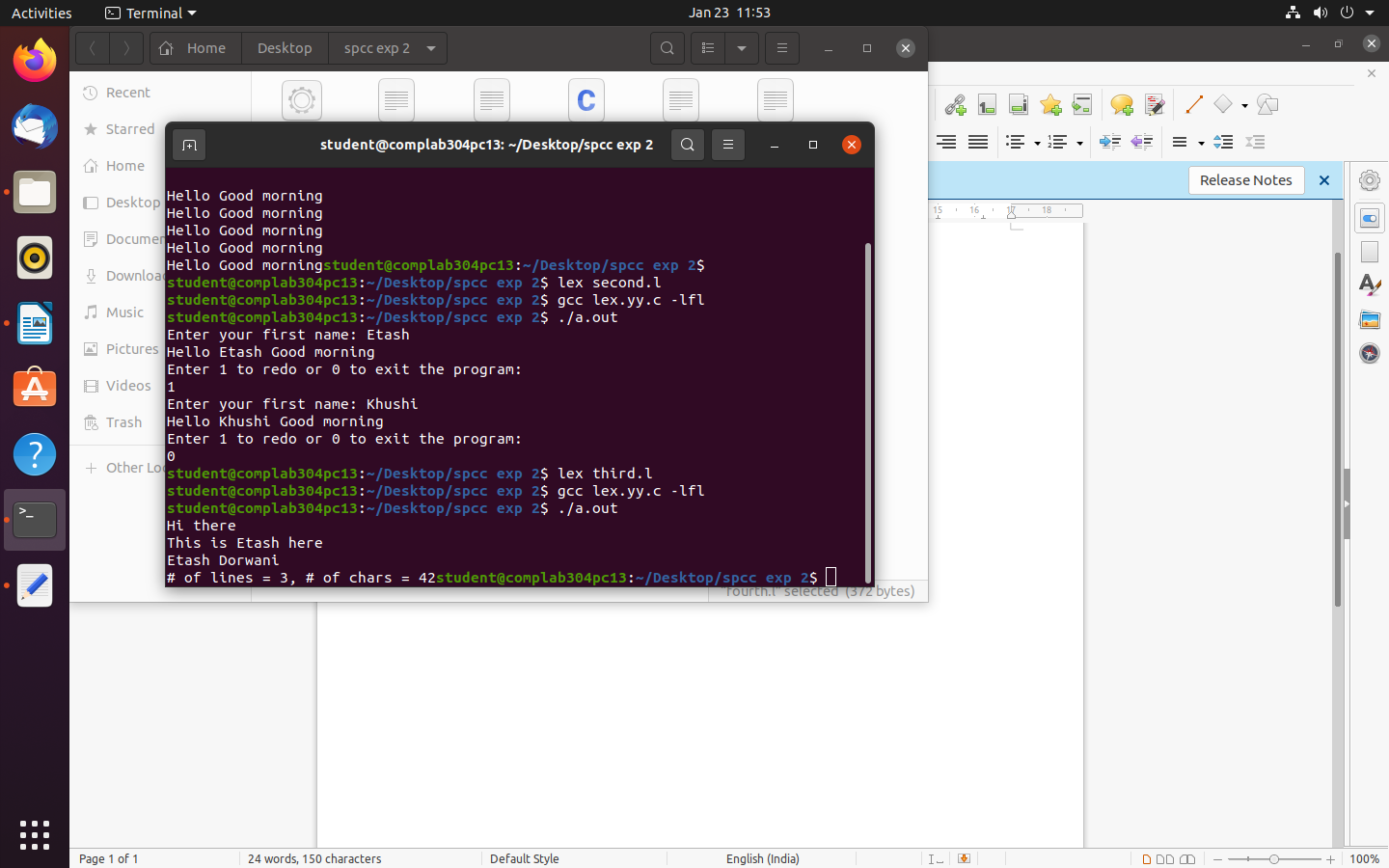
**First.l**



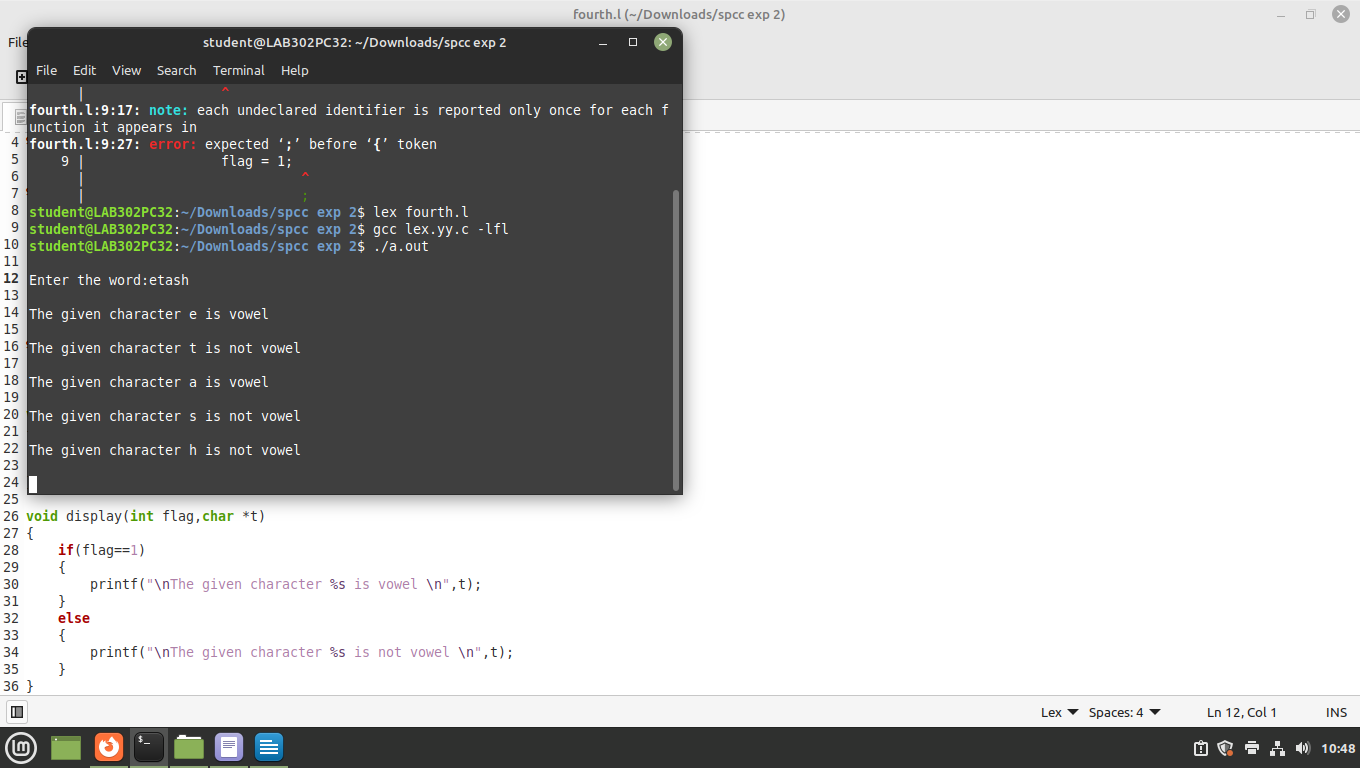
**Second.l**



**Third.l**



**Forth.l**



**Fifth.l**

