NIKIT GOKHE
Class – SY Comp D1
Roll No. 224024
GR No. 21810522

ASSIGNMENT NO 5

AIM:

Write a program using YACC specifications to implement syntax analysis phase of compiler to recognize simple and compound sentences given in input file.

THEORY:

Yacc takes a concise description of a grammar and produces a C routine that can parse that grammar, a parser. The yacc parser automatically detects whenever a sequence of input tokens matches one of the rules in the grammar and also detects a syntax error whenever its input doesn't match any of the rules. A yacc parser is generally not as fast as a parser you could write by hand, but the ease in writing and modifying the parser is invariably worth any speed loss. The amount of time a program spends in a parser is rarely enough to be an issue anyway.

There are mainly three kinds of sentences in English: simple, complex and compound.

Simple sentence :

A simple sentence consists of just one clause. Examples are given below.

- The dog barks.
- The kettle boils.
- Birds live in nests.
- The boys are singing.

In its simplest form, a simple sentence consists of a subject and a verb. We can add more meaning to the sentence by including qualifiers, objects, complements etc.

- Sitting on a branch, the monkey gibbered.
- The little girl was carrying a basket on her head.

Compound sentence : A compound sentence is made up of two or more independent clauses.

- The boys sang and the girls danced.

This compound sentence consists of two simple clauses connected by the coordinating con- junction and.

Another example is given below.

- Men may come and men may go, but I go on forever.

This compound sentence consists of three independent clauses.

We make compound sentences by joining independent clauses with the help of coordinating conjunctions.

More examples of compound sentences are given below

- You may watch TV or you can go out to play.
- You should either sit quietly or go out.
- The old man could neither see nor hear.
- He took the test several times but he couldn't pass.

This program tells whether the input given is compound sentence or simple one. Its easy, just check the scanned symbol with some conjunctives and if they are found, then its compound sentence. Else a simple one. We use a flag variable in this code which is initially 0 indicating a simple sentence. If any conjunctives are matched, set the flag to 1. at the end, if flag=1 then compound sentence, else simple sentence.

Algorithm:

1. Start
2. Create a lex file.
3. Define the rules in the lex file your simple and compound sentences
4. Create a yacc file with same name as lex file.
5. Define tokens like Subject Verb Object and Conjunction.

6. Start with start:Simple/Compound as input can be simple or compound sentence.
7. Define label Simple and define its formation as Simple:Subject Verb Object.
8. If given sentence is as simple sentence the print result as ”Simple Sentence”.
9. Define label Compound and define its formation as Compound:Simple Conjunction Simple.
10. If given sentence is as simple sentence the print result as ”Compound Sentence”.
11. In main() call function yyparse()and yywrap()
12. End

Input to the program: The boys are singing

Output: Simple Sentence

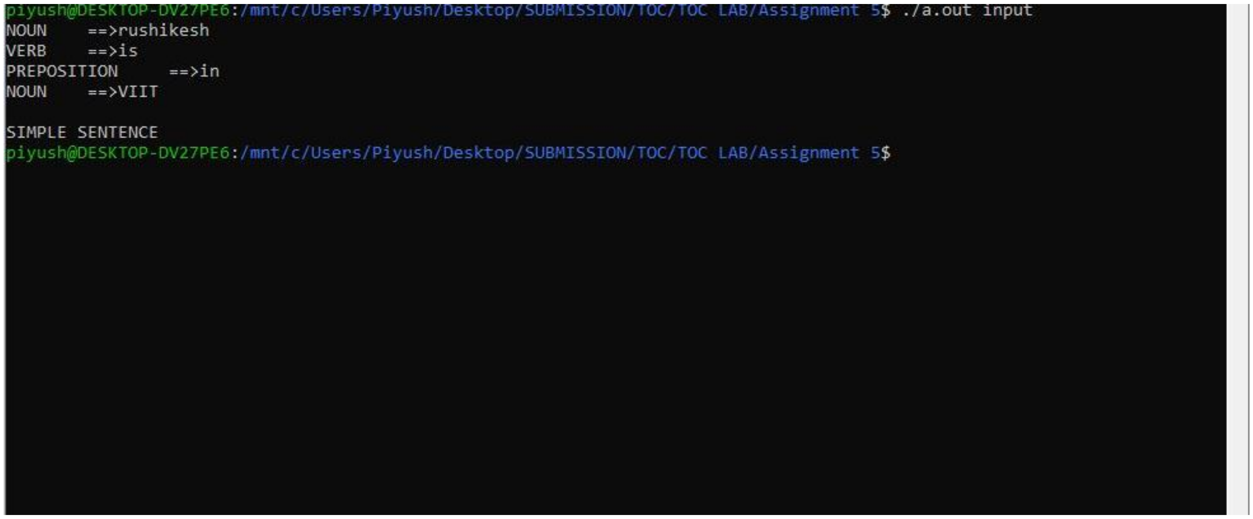
LEX CODE:

```
%{
#include "y.tab.h" //Contains Token Definiation
%}
%%
[\t ] ;     //IGNORE WHITE SPACES
am|is|are|have|has|can|will|shall|eat|sing|go|goes { printf("VERB\t==>%s\n",yytext);return
VERB;}
very|simply|gently { printf("VERB\t==>%s\n",yytext);return(ADVERB); }
and|or|also|so|but|if|then {printf("CONJUNCTION\t==>%s\n",yytext);return
(CONJUNCTION);}
fast|good|honest {printf("ADJECTIVE\t==>%s\n",yytext);return (ADJECTIVE);}
I|he|she|we|they|you|this {printf("PRONOUN\t==>%s\n",yytext);return (PRONOUN);}
in|on|to {printf("PREPOSITION\t==>%s\n",yytext);return (PREPOSITION);}
[a-zA-Z]+ {printf("NOUN\t==>%s\n",yytext);return (NOUN);}
. ; //IGNORE ANYTHING ELSE
%%
int yywrap()
{
return 1;
}
```

YACC CODE:

```
%{
#include<stdio.h>
void yyerror(char*);
int yylex();
FILE* yyin;
%}
%token NOUN PRONOUN ADJECTIVE VERB ADVERB CONJUNCTION PREPOSITION
%%
sentence: compound { printf("COMPOUND SENTENCE\n");}
            |
            simple {printf("SIMPLE SENTENCE\n");}
            ;
simple: subject VERB object;
compound: subject VERB object CONJUNCTION subject VERB object;
subject: NOUN|PRONOUN;
object: NOUN|ADJECTIVE NOUN|ADVERB NOUN|PREPOSITION NOUN;
%%
void yyerror(char *s)
{
printf("ERROR:%s",s);
}
int main(int argc,char* argv[])
{
yyin=fopen(argv[1],"r");
yyparse();
fclose(yyin);
return 0;
}
```

OUTPUT :



```
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 5$ ./a.out input
NOUN    ==>rushikesh
VERB    ==>is
PREPOSITION     ==>in
NOUN    ==>VIIT

SIMPLE SENTENCE
piyush@DESKTOP-DV27PE6:/mnt/c/Users/Piyush/Desktop/SUBMISSION/TOC/TOC LAB/Assignment 5$
```