Experiment 6

Lex Yacc Parser

Q1

Grammar

G->AGBB | epsilon

modified grammar:

G->AGBB | epsilon

S->G

Code

.y

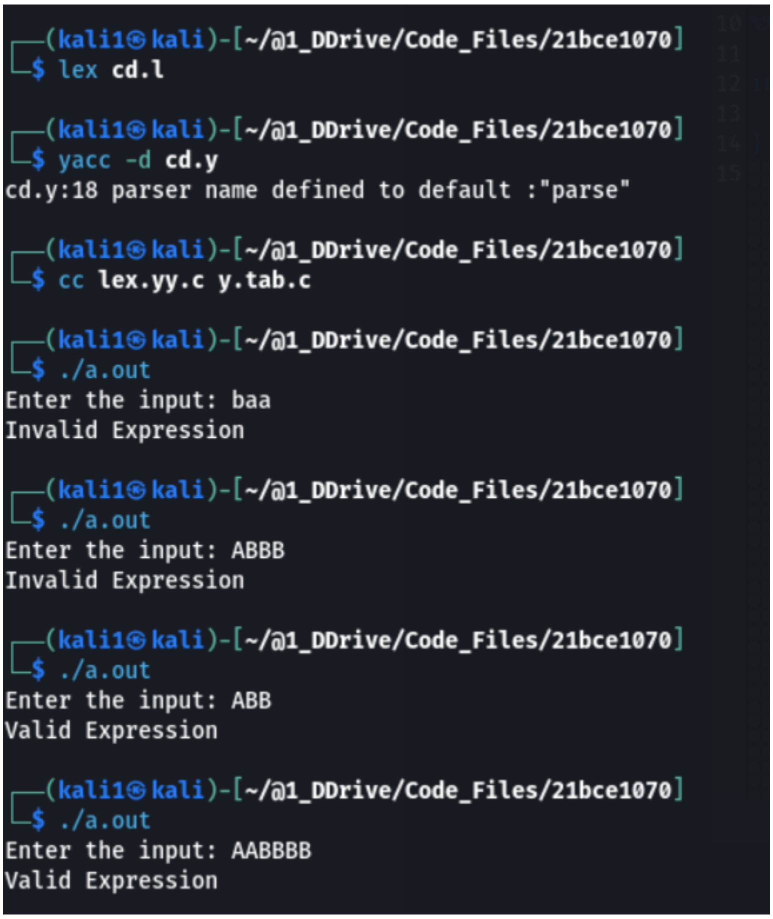
```
%{
#include <stdio.h>
int yylex(void);
int yyerror(char *s);
%}

%token A B

%%
S: G { printf("Valid Expression\n"); }
G: A G B B
|
;
%%
int main() {
   printf("Enter the input: ");
   yyparse();
```

```
    return 0;
}

int yyerror(char *s) {
    printf("Invalid Expression\n");
    return 0;
}
```

.l

```
%{
#include "y.tab.h"
%}

%%
A { return A; }
B { return B; }
\n { return 0; } //end this .l code loop when \n is input
. { return yytext[0]; }
%%

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

Output



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ lex cd.l

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ yacc -d cd.y
cd.y:18 parser name defined to default :"parse"

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ cc lex.yy.c y.tab.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: baa
Invalid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: ABBB
Invalid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: ABB
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: AABBBB
Valid Expression
```

Q2

Code

.y

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

int yylex(void);
int yyerror(char *s);
%}

%token STRING
```

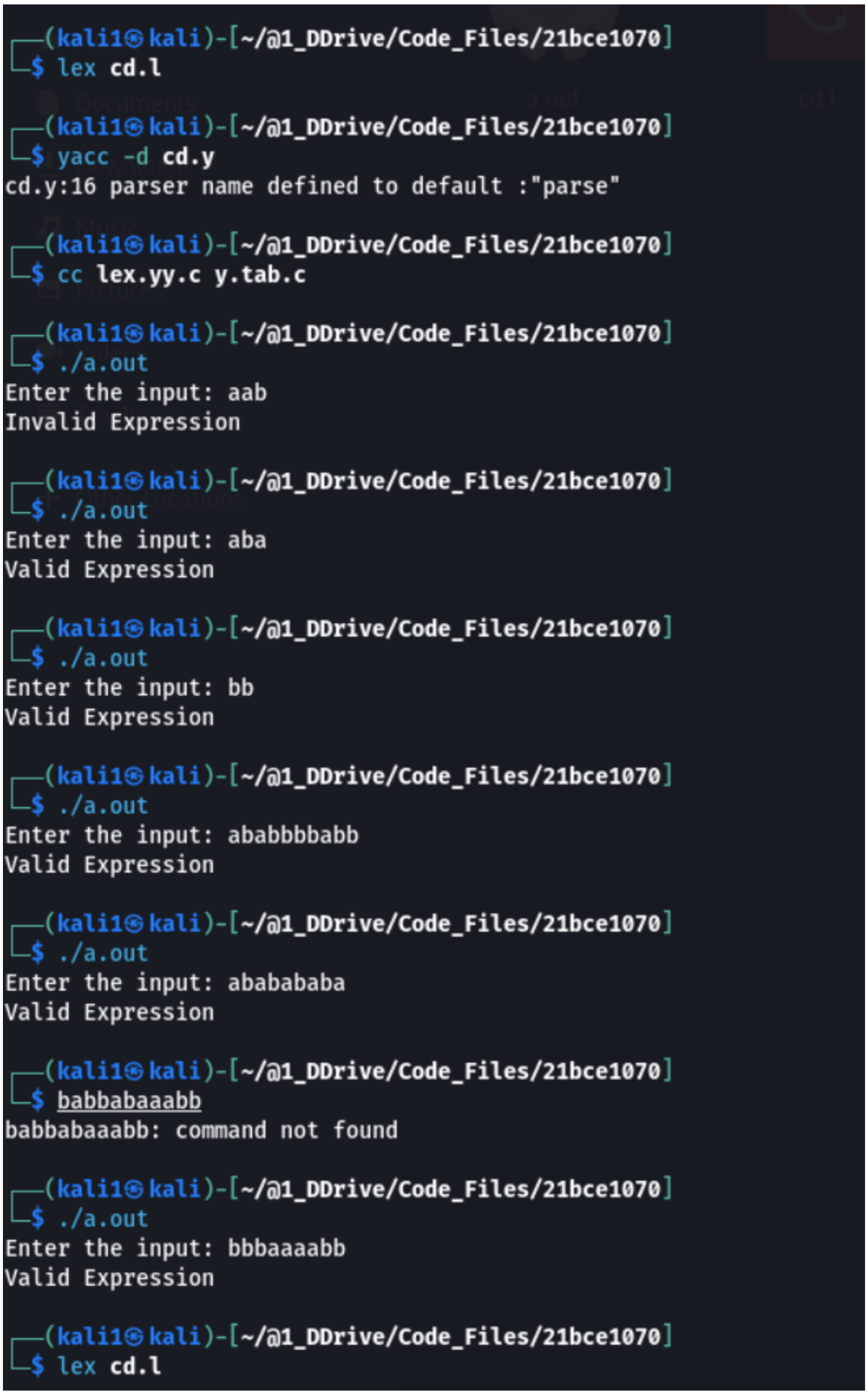
```
%%

S: STRING { printf("Valid Expression\n"); }
;

%%

int main() {
    printf("Enter the input: ");
    yyparse();
    return 0;
}

int yyerror(char *s) {
    printf("Invalid Expression\n");
    return 0;
}
```

.l

```
%{
#include <stdio.h>
#include "y.tab.h"
%}

%%

"aba".*   { return STRING; }
.*"bb"    { return STRING; }
\n       { return 0; }
```

```
.        { return yytext[0]; }


%%


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

Output



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ lex cd.l

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ yacc -d cd.y
cd.y:16 parser name defined to default :"parse"

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ cc lex.yy.c y.tab.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: aab
Invalid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: aba
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: bb
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: ababbbbabb
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: abababababa
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ babbabaaabb
babbabaaabb: command not found

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: bbbaaaabb
Valid Expression

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ lex cd.l
```

Q3

Code

.l

```
%{
```

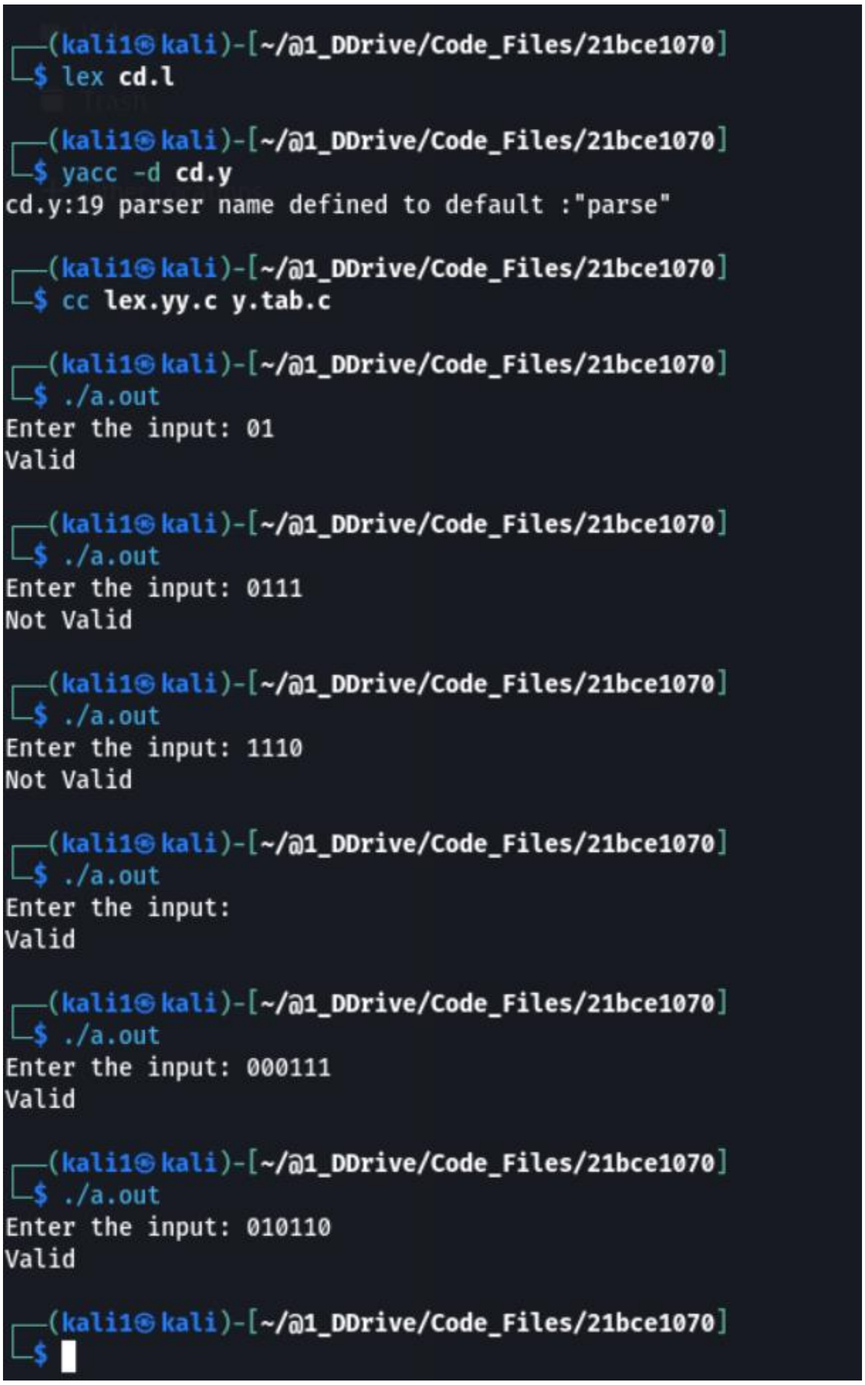
```
#include "y.tab.h"
%}

%%
"0" { return A; }
"1" { return B; }
\n { return 0; } //end this .l code loop when \n is input
. { return yytext[0]; }
%%

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

.y

```
%{
#include <stdio.h>
int yylex(void);
int yyerror(char *s);
int c0=0,c1=0;
%}

%token A B



%%
S: G { if (c1==c0) printf("Valid"); else printf("Not Valid");}
```

```
G: A G {c1++;}
|B G {c0++;}
|
;
%%




int main() {
    printf("Enter the input: ");
    yyparse();
    return 0;
}

int yyerror(char *s) {
    printf("Invalid Expression\n");
    return 0;
}
```

Output



```
┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ lex cd.l

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ yacc -d cd.y
cd.y:19 parser name defined to default :"parse"

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ cc lex.yy.c y.tab.c

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: 01
Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: 0111
Not Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: 1110
Not Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input:
Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: 000111
Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ ./a.out
Enter the input: 010110
Valid

┌──(kali1㉿kali)-[~/@1_DDrive/Code_Files/21bce1070]
└─$ █
```