

Siddhant Bhagat
22BCE0682

SAMPLE CODE:
LEX CODE:

```
*St.l
1 %{
2 #include<stdio.h>
3 #include"Sty.tab.h"
4 %}
5 L      [a-zA-Z]
6 D      [0-9]
7 ID     {L}({L}|{D})*
8 NUM    {D}+
9 AO     [-+*/]ss|
10 PO     [()]
11 ASO    =
12 SO     [;\n]
13 %%
14 {ID}   return(ID);
15 {NUM}  return(NUM);
16 {AO}   return *yytext;
17 {PO}   return *yytext;
18 {ASO}  return *yytext;
19 {SO}   return *yytext;
20 %%
```

YAK CODE:

```
*St.l      *St.y
1 |
2 #include<stdio.h>
3 #include"Sty.tab.h"
4 void yyerror(char *);
5 int yylex(void);
6 |
7 #token ID NUM
8 %
9 S      : ID '=' E ';' '\n'      printf("Valid Statement\n");
10 E      : E '+' T
11         | E '-' T
12         | T
13 T      : T '*' F
14         | T '/' F
15         | F
16 F      : '(' E ')'
17         | ID
18         | NUM
19 %%
20 int main()
21 {
22     printf("Enter the statement:");
23     yyparse();
24     return 0;
25 }
26
27 int yywrap()
28 {
29     return 1;
30 }
31
32 void yyerror(char *err)
33 {
34     fprintf(stderr,"%s",err);
35 }
```

OUTPUT

```
matlab@sjt318scope060:~$ flex St.l
matlab@sjt318scope060:~$ bison -d Sty.y
matlab@sjt318scope060:~$ gcc lex.yy.c Sty.tab.c
matlab@sjt318scope060:~$ ./a.out
Enter the statement:A=1+2;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:E=E+T;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:E=E-T;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:E=T;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:T=T*F;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:T=T/F;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:T=F;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:F=(E);
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:F=ID;
Valid Statement

syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:F=NUM;
Valid Statement

matlab@sjt318scope060:~$ █
```

Q1)

LEX CODE:

```
1 %{
2 #include<stdio.h>
3 #include"Sty.tab.h"
4
5 %}
6
7 ID      [a-d]
8 SO      [\n]
9
10 %%
11 {ID}    return(ID);
12 {SO}    return *yytext;
13 %|
```

YAK Code:

```

1 %{
2 #include<stdio.h>
3 #include"Sty.tab.h"
4 void yyerror(char *);
5 int yylex(void);
6 %}
7 %token ID
8 %%
9 S      : ID A B ID      {printf("Valid Statement\n");}
10 A     : ID
11       | ' '
12 B     : ID
13       | ' '
14 %%
15 int main()
16 {
17     printf("Enter the statement:");
18     yyparse();
19     return 0;
20 }
21
22 int yywrap()
23 {
24     return 1;
25 }
26
27 void yyerror(char *err)
28 {
29     fprintf(stderr,"%s",err);
30 }

```

OUTPUT:

```

syntax errormatlab@sjt318scope060:~$ flex St.l
matlab@sjt318scope060:~$ bison -d Sty.y
matlab@sjt318scope060:~$ gcc lex.yy.c Sty.tab.c
matlab@sjt318scope060:~$ ./a.out
Enter the statement:abcd
Valid Statement
matlab@sjt318scope060:~$ 

```

Q2)

LEX CODE:

```
St.l
1 %{
2 #include<stdio.h>
3 #include"Sty.tab.h"
4 %}
5 L      [a-zA-Z]
6 CH     [a b 5 3]
7 AO     [-+*/]
8 PO     [()]
9 ASO    [=]
10 SO    [;\n]
11 %%
12 {CH}   return *yytext;
13 {AO}   return *yytext;
14 {PO}   return *yytext;
15 {ASO}  return *yytext;
16 {SO}   return *yytext;
17 %%
```

YAK CODE:

```

1
2 #include<stdio.h>
3 #include"Sty.tab.h"
4 void yyerror(char *);
5 int yylex(void);
6
7 %token ID NUM
8 %%
9 S      : 'a' '=' E ';' '\n'      {printf("Valid Statement\n");}
10 E      : '5' '+' T
11         | '3' '-' T
12 T      : V
13         | V '*' V
14         | V '+' V
15 V      : 'a'
16         | 'b'
17 %%
18 int main()
19 {
20     printf("Enter the statement:");
21     yyparse();
22     return 0;
23 }
24
25 int yywrap()
26 {
27     return 1;
28 }
29
30 void yyerror(char *err)
31 {
32     fprintf(stderr,"%s",err);
33 }

```

OUTPUT:

```

matlab@sjt318scope060:~$ bison -d Sty.y
matlab@sjt318scope060:~$ gcc lex.yy.c Sty.tab.c
matlab@sjt318scope060:~$ ./a.out
Enter the statement:V=A;
syntax errorVmatlab@sjt318scope060:~$ ./a.out
Enter the statement:5-3;
syntax errormatlab@sjt318scope060:~$ ./a.out
Enter the statement:3-a
syntax errormatlab@sjt318scope060:~$
matlab@sjt318scope060:~$ lex St.l
matlab@sjt318scope060:~$ bison -d Sty.y
matlab@sjt318scope060:~$ gcc lex.yy.c Sty.tab.c
matlab@sjt318scope060:~$ ./a.out
Enter the statement:V=3-a;
syntax errorVmatlab@sjt318scope060:~$ ./a.out
Enter the statement:3-a+b;
Valid Statement

```

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

Q4) Write a YACC program to accept the following conditional statement part.

```
for(x=0;x<10;x++)  
{  
x=x+1;  
}
```

22bce0682.1

```
%{  
#include<stdio.h>  
#include"22bce0682.tab.h"  
%}
```

```
L [a-zA-Z]  
D [0-9]  
ID {L}({L}|{D})*  
NUM {D}+  
ROP "<"|">"|"<="|">="|"  
AOP [-+*/]  
PO [0]  
CO [{}]  
AO =  
SO [;\n]  
%%
```

```
if return(IF);  
while return(WHILE);  
for return(FOR);  
{ID} return(ID);  
{NUM} return(NUM);  
{ROP} return(ROP);  
{AOP} return *yytext;  
{PO} return *yytext;  
{CO} return *yytext;  
{AO} return *yytext;  
{SO} return *yytext;  
%%
```

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

```
22bce0682.l x 22bce0682.y
home > matlab > 22bce0682.l
1  %{
2  #include<stdio.h>
3  #include"22bce0682.tab.h"
4  %}
5
6  L [a-zA-Z]
7  D [0-9]
8  ID {L}({L}|{D})*
9  NUM {D}+
10 ROP "<" | ">" | "<=" | ">="
11 AOP [-+*/]
12 PO [(())]
13 CO [{}]
14 AO =
15 SO [;\n]
16 %%
17
18 if return(IF);
19 while return(WHILE);
20 for return(FOR);
21 {ID} return(ID);
22 {NUM} return(NUM);
23 {ROP} return(ROP);
24 {AOP} return *yytext;
25 {PO} return *yytext;
26 {CO} return *yytext;
27 {AO} return *yytext;
28 {SO} return *yytext;
29 %%
30
```

22bce0682.y

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

```
%{
#include<stdio.h>
#include"22bce0682.tab.h"
int yylex(void);
void yyerror(char *);
}%

%token IF WHILE FOR NUM ID ROP INC

%%
S :IF('C'){ID='E';}"\n" {printf("Valid Statement\n");}
  |WHILE('C'){ID='E';}"\n" {printf("Valid Statement\n");}
  |FOR('C'='C';'C';ID+"+"){ID='E';}"\n" {printf("Valid Statement\n");}
C :IN ROP IN
  |ID
  |NUM
E :E+'T
  |E-'T
  |T
T :T'*'F
  |T/'F
  |F
F :('E')
  |ID
  |NUM
IN: ID
  |NUM

%%
int main()
{
printf("Enter statement : ");
yyparse();
return 0;
}

int yywrap()
{
return 1;
}

void yyerror(char *s)
{
fprintf(stderr, "%s\n", s);
}
```


Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

22bce0682.l

22bce0682.y x

home > matlab > 22bce0682.y

```
1  %{
2  #include<stdio.h>
3  #include"22bce0682.tab.h"
4  int yylex(void);
5  void yyerror(char *);
6  %}
7
8  %token IF WHILE FOR NUM ID ROP INC
9
10 %%
11 S :IF '('C')' '{'ID'='E';'}'\n' {printf("Valid Stat
12   |WHILE '('C')' '{'ID'='E';'}'\n' {printf("Valid
13   |FOR '('C'='C';'C';'ID'+'+'')' '{'ID'='E';'}'\n
14 C :IN ROP IN
15   |ID
16   |NUM
17 E :E '+'T
18   |E '-'T
19   |T
20 T :T '*'F
21   |T '/'F
22   |F
23 F : '('E')'
24   |ID
25   |NUM
26 IN: ID
27   |NUM
28
```

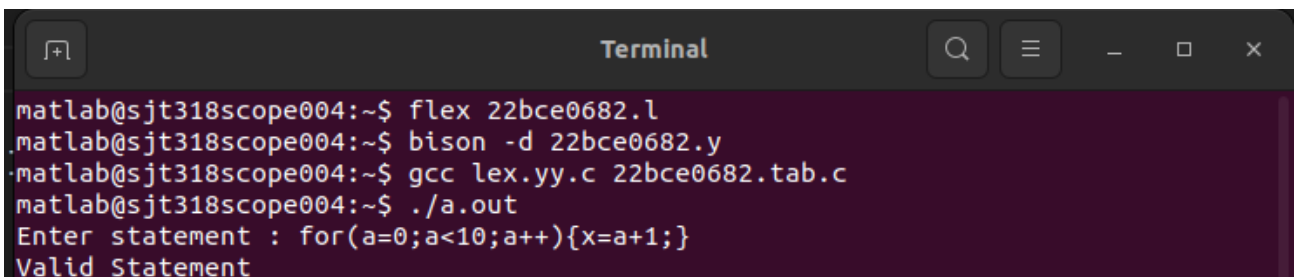
Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

```
28
29
30 %%
31 int main()
32 {
33     printf("Enter statement : ");
34     yyparse();
35     return 0;
36 }
37
38 int yywrap()
39 {
40     return 1;
41 }
42
43 void yyerror(char *s)
44 {
45     fprintf(stderr, "%s\n", s);
46 }
47
```

OUTPUT:



```
matlab@sjt318scope004:~$ flex 22bce0682.l
matlab@sjt318scope004:~$ bison -d 22bce0682.y
matlab@sjt318scope004:~$ gcc lex.yy.c 22bce0682.tab.c
matlab@sjt318scope004:~$ ./a.out
Enter statement : for(a=0;a<10;a++){x=a+1;}
Valid Statement
```

Q3) Write a YACC program to accept the following conditional statement part.

```
if (a>=b){
x = a+b;
}
```

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

Lex program

```
1 %{
2 #include<stdio.h>
3 #include"statey.tab.h"
4 %}
5
6 L [a-zA-Z]
7 D [0-9]
8 ID {L}({L}|{D})*
9 NUM {D}+
10 ROP "<"| ">"| "<="| ">="
11 AOP [-+*/]
12 PO [( )]
13 CO [{ } ]
14 AO =
15 SO [;\n]
16 %%
17
18 if return(IF);
19 while return(WHILE);
20 for return(FOR);
21 {ID} return(ID);
22 {NUM} return(NUM);
23 {ROP} return(ROP);
24 {AOP} return *yytext;
25 {PO} return *yytext;
26 {CO} return *yytext;
27 {AO} return *yytext;
28 {SO} return *yytext;
29 %%
30
```

Yacc Program

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

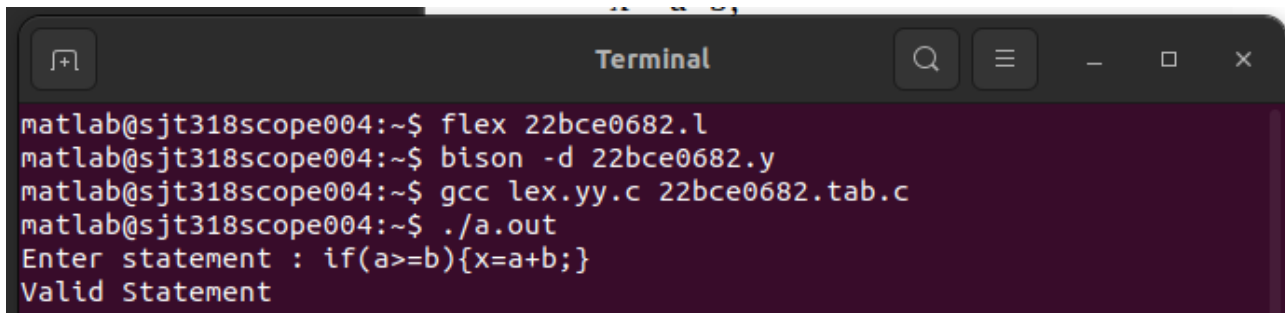
```
1 %{
2 #include<stdio.h>
3 #include"statey.tab.h"
4 int yylex(void);
5 void yyerror(char *);
6 %}
7
8 %token IF WHILE FOR NUM ID ROP INC
9
10 %%
11 S :IF('C')''{'ID'='E';''}\n' {printf("Valid Statement\n");}
12   |WHILE('C')''{'ID'='E';''}\n' {printf("Valid Statement\n");}
13   |FOR('C'='C';'C';'ID'+'+'')''{'ID'='E';''}\n' {printf("Valid Statement\n");}
14 C :IN ROP IN
15   |ID
16   |NUM
17 E :E+'T
18   |E-'T
19   |T
20 T :T'*'F
21   |T/'F
22   |F
23 F : '('E')'
24   |ID
25   |NUM
26 IN: ID
27   |NUM
28
29
30 %%
31 int main()
32 {
33     printf("Enter statement : ");
34     yyparse();
35     return 0;
36 }
37
38 int yywrap()
39 {
40     return 1;
41 }
42
43 void yyerror(char *s)
44 {
45     fprintf(stderr, "%s\n", s);
46 }
47
```

OUTPUT:

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682



```
matlab@sjt318scope004:~$ flex 22bce0682.l
matlab@sjt318scope004:~$ bison -d 22bce0682.y
matlab@sjt318scope004:~$ gcc lex.yy.c 22bce0682.tab.c
matlab@sjt318scope004:~$ ./a.out
Enter statement : if(a>=b){x=a+b;}
Valid Statement
```

Q1)

Write a YACC program to implement the following grammar and check the implemented grammar using two test cases.

$S \rightarrow E;$

$E \rightarrow E-T|E+T|T$

$T \rightarrow T*F|T/F|F$

$F \rightarrow (E) | ID | Num$

LEX program:

```
%{
#include "22BCE0682.tab.h"
%}

%%

[a-zA-Z]+ { return ID; }
[0-9]+    { yylval = atoi(yytext); return NUM; }
"+"      { return '+'; }
"-"      { return '-'; }
"*"      { return '*'; }
"/"      { return '/'; }
"("      { return '('; }
")"      { return ')'; }
";"      { return ';'; }

[ \t\n]+ { }

.        { return yytext[0]; }

%%

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

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

YACC program:

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

int yylex(void);
void yyerror(const char*);

%}

%token ID NUM
%left '+' '-'
%left '*' '/'
%right UMINUS

%%

S : E ';' { printf("Valid statement\n"); }
  ;

E : E '+' T { printf("Adding\n"); }
  | E '-' T { printf("Subtracting\n"); }
  | T
  ;

T : T '*' F { printf("Multiplying\n"); }
  | T '/' F { printf("Dividing\n"); }
  | F
  ;
```

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

```
F : '(' E ')' { printf("Expression inside parentheses\n"); }
| ID      { printf("Identifier\n"); }
| NUM     { printf("Number\n"); }
;

%%

void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}

int main() {
    printf("Enter expression followed by a semicolon:\n");
    return yyparse(); // Start the parsing process
}
```

OUTPUT:

```
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ flex 22BCE0682.l
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ bison -d 22BCE0682.y
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ gcc lex.yy.c 22BCE0682.tab.c
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ ./a.out
Enter expression followed by a semicolon:
5/(3+2)*4;
Number
Number
Number
Adding
Expression inside parentheses
Dividing
Number
Multiplying
Valid statement
```

Q2)

Write a YACC program to implement the following grammar and check using the two test cases acdb and adb.

S -> aABb

A -> c | €

B -> d | €

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

LEX Code:

```
%{  
#include "22BCE0682sid.tab.h"  
%}  
  
%%  
  
a  { return 'a'; }  
b  { return 'b'; }  
c  { return 'c'; }  
d  { return 'd'; }  
  
[ \t\n]+ { }  
  
.  { return yytext[0]; }  
  
%%  
  
int yywrap() {  
    return 1;  
}
```

YACC program:

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

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

// Function declarations
int yylex(void);
void yyerror(const char *s);

%}

%token a b c d

%%

S : 'a' A B 'b' { printf("Valid string\n"); }
  ;

A : 'c'          { printf("A -> c\n"); }
  |              { printf("A -> ε (epsilon)\n"); }
  ;

B : 'd'          { printf("B -> d\n"); }
  |              { printf("B -> ε (epsilon)\n"); }
  ;

%%

void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}

int main() {
    printf("Enter a string to parse: ");
    return yyparse();
}
```

OUTPUT:

Digital Lab Assessment-3

Siddhant Bhagat

22BCE0682

```
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ flex 22BCE0682sid.l
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ bison -d 22BCE0682sid.y
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ gcc lex.yy.c 22BCE0682sid.tab.c
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ ./a.out
Enter a string to parse: adb
A -> ε (epsilon)
B -> d
Valid string
```

```
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ flex 22BCE0682sid.l
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ bison -d 22BCE0682sid.y
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ gcc lex.yy.c 22BCE0682sid.tab.c
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ flex 22BCE0682sid.l
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ bison -d 22BCE0682sid.y
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ gcc lex.yy.c 22BCE0682sid.tab.c
siddhant-bhagat@siddhant-bhagat-HP-Spectre-x360-2-in-1-Laptop-14-ef0xxx:~/Desktop$ ./a.out
Enter a string to parse: acdb
A -> c
B -> d
Valid string
```