

NAME: VAIBHAV BANKA

REG NO: 21BCE1955

EXERCISE 4

1. Construct Yacc program for the grammar

E -> E*T | E+T | T

T -> T-T | F

F -> NUM

LEX PROGRAM

```
%{  
    #include "y.tab.h"  
    extern int yylval;  
%}  
%%  
[0-9]+ {  
    yylval = atoi(yytext);  
    return NUMBER;  
}  
[\t]+ ;  
\n {return 0;}  
. {return yytext[0];}  
%%  
int yywrap(){}
```

YACC PROGRAM

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

```
%token NUMBER
```

```
%left '+'*
```

```
%left '-'
```

```
%%
```

```
S : E {
```

```
    printf("Result = %d\n", $$);
```

```
    return 0;}
```

```
E :
```

```
    E'*'T {$$ = $1*$3;}
```

```
    |E'+'T {$$ = $1+$3;}
```

```
    |T {$$ = $1;}
```

```
T :
```

```
    T'-'T {$$ = $1-$3;}
```

```
    |F {$$ = $1;}
```

```
F : NUMBER{$$ = $1;}
```

```
%%
```

```
int main(){
```

```
    printf("enter the expression\n");
```

```
    yyparse();
```

```
}
```

```
int yyerror(char* s){
```

```
    printf("\nExpression is invalid\n");
```

```
}
```

OUTPUT

```

vaibhav@vaibhav-virtual-machine:~$ lex ex4_1.l
vaibhav@vaibhav-virtual-machine:~$ yacc -d ex4_1.y
ex4_1.y:27 parser name defined to default : "parse"
vaibhav@vaibhav-virtual-machine:~$ cc lex.yy.c y.tab.c
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the expression
7+9
Result = 16
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the expression
7*9
Result = 63
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the expression
5+7+6
Result = 18
vaibhav@vaibhav-virtual-machine:~$

```

2. $a^n b^n$ $n \geq 0$ $\Sigma = \{a, b\}^*$

LEX PROGRAM

```

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

%%

[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}

%%

int yywrap() {}

```

YACC PROGRAM

```

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

%token A B NL

```

%%

```
stmt: S NL { printf("valid string\n");  
          exit(0); }
```

;

S: A S B |

;

%%

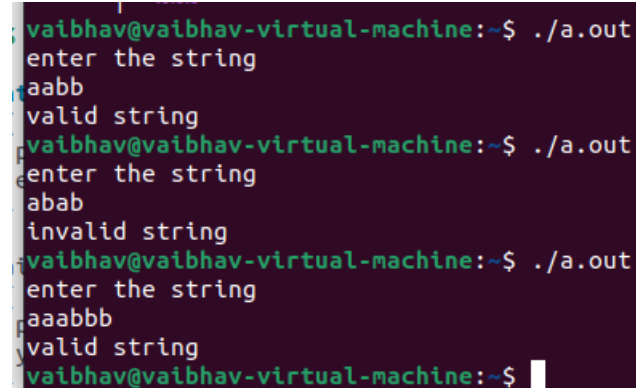
```
int yyerror(char *msg)
```

```
{  
    printf("invalid string\n");  
    exit(0);  
}
```

```
main()
```

```
{  
    printf("enter the string\n");  
    yyparse();  
}
```

OUTPUT



```
vaibhav@vaibhav-virtual-machine:~$ ./a.out  
enter the string  
aabb  
valid string  
vaibhav@vaibhav-virtual-machine:~$ ./a.out  
enter the string  
abab  
invalid string  
vaibhav@vaibhav-virtual-machine:~$ ./a.out  
enter the string  
aaabbb  
valid string  
vaibhav@vaibhav-virtual-machine:~$
```

3. $a^n b^n$ $n \geq 3$ $\Sigma = \{a, b\}^*$

LEX PROGRAM

%{

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

%}

%%

[aA] {return A;}
[bB] {return B;}
. {return yytext[0];}

%%
```

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

YACC PROGRAM

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

```
%token A B

%%

stmt: S { printf("valid string\n");
        exit(0); }
;
S: A S B | A A B B B
;

%%
```

```
int yyerror(char *msg)
{
    printf("invalid string\n");
}
```

```

    exit(0);
}

main()
{
    printf("enter the string\n");
    yyparse();
}

```

OUTPUT

```

t vaibhav@vaibhav-virtual-machine:~$ ./a.out
s enter the string
t aaabbb
s valid string
: vaibhav@vaibhav-virtual-machine:~$ ./a.out
: enter the string
s aabb
s invalid string
n vaibhav@vaibhav-virtual-machine:~$ ./a.out
n enter the string
t ab
t invalid string
} vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
a aaaabbbb
{ valid string
{ vaibhav@vaibhav-virtual-machine:~$

```

4. Check whether the string of alphabet a,b is palindrome or not $\Sigma = \{a,b\}^*$

LEX PROGRAM

```

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

%%

[aA] {return A;}
[bB] {return B;}
\n {return NL;}
. {return yytext[0];}

```

```
%%
```

```
int yywrap() {}
```

YACC PROGRAM

```
%{
```

```
    #include<stdio.h>
```

```
    #include<stdlib.h>
```

```
%}
```

```
%token A B NL
```

```
%%
```

```
stmt: S NL { printf("valid string\n");  
            exit(0); }
```

```
;
```

```
S: A S A
```

```
    | B S B
```

```
    | A | B |
```

```
;
```

```
%%
```

```
int main()
```

```
{
```

```
    printf("enter the string\n");
```

```
    yyparse();
```

```
    return 0;
```

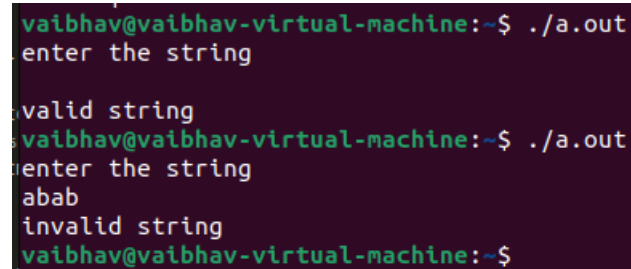
```
}
```

```
int yyerror(char *msg)
```

```
{
```

```
printf("invalid string\n");  
}
```

OUTPUT



```
vaibhav@vaibhav-virtual-machine:~$ ./a.out  
enter the string  
valid string  
vaibhav@vaibhav-virtual-machine:~$ ./a.out  
enter the string  
abab  
invalid string  
vaibhav@vaibhav-virtual-machine:~$
```

5. All strings starting with 0 and ending with 1 $\Sigma = \{0,1\}^*$

LEX PROGRAM

```
%{  
    #include "y.tab.h"  
}%  
  
%%  
  
[0] {return ZERO;}  
[1] {return ONE;}  
\n {return NL;}  
. {return yytext[0];}  
%%  
  
int yywrap() {}
```

YACC PROGRAM

```
%{  
    #include<stdio.h>  
    #include<stdlib.h>  
}%  
  
%token ZERO ONE NL  
%%
```



```

stmt: S NL { printf("valid string\n");
        exit(0); }

;

S: N | ZERO A

;

A: N A | ONE

;

N: ZERO | ONE

;

%%

```

```

int yyerror(char *msg)
{
    printf("invalid string\n");
    exit(0);
}

main()
{
    printf("enter the string\n");
    yyparse();
}

```

OUTPUT

```

vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
01
valid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
0111
valid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
101
invalid string
vaibhav@vaibhav-virtual-machine:~$

```

6. All possible strings starting with 01 and ending with 10 $\Sigma = \{0,1\}^*$

LEX PROGRAM

```
%{  
    #include "y.tab.h"  
%}  
  
%%  
  
[0] {return ZERO;}  
[1] {return ONE;}  
\n {return NL;}  
. {return yytext[0];}  
%%
```

```
int yywrap() {}
```

YACC PROGRAM

```
%{  
    #include<stdio.h>  
    #include<stdlib.h>  
%}  
  
%token ZERO ONE NL  
%%  
  
stmt: S NL { printf("valid string\n");  
            exit(0); }  
  
;  
  
S: ZERO B  
  
;  
  
B: ONE C  
  
;
```

C: ONE C | ZERO D

;

D: ZERO E | ONE C |

;

E: ZERO E | ONE C

;

%%

```
int yyerror(char *msg)
```

```
{
```

```
    printf("invalid string\n");
```

```
    exit(0);
```

```
}
```

```
main()
```

```
{
```

```
    printf("enter the string\n");
```

```
    yyparse();
```

```
}
```

OUTPUT

```

vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
010
valid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
0110
valid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
0111
invalid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
010
valid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
011101010110101010110101010100110
valid string
vaibhav@vaibhav-virtual-machine:~$ █

```

7. Equal number of 0's and 1's $\Sigma = \{0,1\}^*$

LEX PROGRAM

```

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

%%

[a] {return A;}
[b] {return B;}
\n {return NL;}
. {return yytext[0];}

%%

int yywrap() {}

```

YACC PROGRAM

```

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

```

```
int yylex(void);

int yyerror(char* s);

%}

%token A B NL

%left A B

%%
```

```
stmt: S {printf("accepted\n");exit(0);}

;

S: S S

| A S B

| B S A

| A B

| B A

|

;

%%
```

```
int main()

{

    printf("enter the string\n");

    yyparse();

    return 0;

}

int yyerror(char *msg)

{

    printf("invalid string\n");

}
```

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ cc lex.yy.c y.tab.c
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
abab
accepted
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
baba
accepted
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
babababab
invalid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
abababa
invalid string
vaibhav@vaibhav-virtual-machine:~$ ./a.out
enter the string
abbbbaa
accepted
vaibhav@vaibhav-virtual-machine:~$
```