

Q20. WAP in yacc to evaluate exp of the following grammar:

**$E \rightarrow E+T|T$
 $T \rightarrow T*F|F$
 $F \rightarrow (E)|Digit$**

```
%{
#include<stdio.h>
#include<ctype.h>
}%
%token DIGIT
%left '+'
%left '*'
%%
S:E'\n'{printf("\n%d", $1);}
E:E'+E {$$=$1+$3;}|E'*E {$$=$1*$3;}|('E') {$$=$2;}|DIGIT;
%%
int yylex()
{
    int c=getchar();
    if(isdigit(c))
    {
        yylval=c-'0';
        return DIGIT;
    }
    return c;
}
int yyerror()
{
    return 1;
}
int yywrap()
{
    return 1;
}
int main()
{
    yyparse();
    printf("\n");
    return 1;
}
```

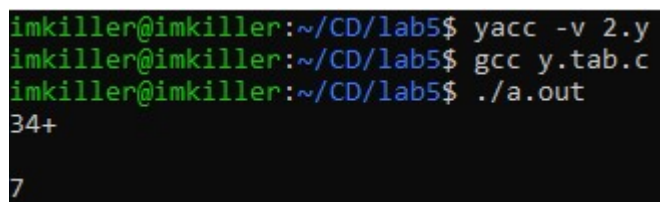
OUTPUT:

```
imkiller@imkiller:~/CD/lab5$ yacc -v 1.y
imkiller@imkiller:~/CD/lab5$ gcc y.tab.c
imkiller@imkiller:~/CD/lab5$ ./a.out
4*4+6
22
```

Q21. WAP in yacc to calculate postfix expression:

```
%{
#include<stdio.h>
#include<ctype.h>
%}
%token num
%left '+' '-'
%left '*' '/'
%right '^'
%%
s:e^n'{printf("\n%d", $1);}
e:e e+'{'$$=$1+$2;}|e e-'{'$$=$1-$2;}|e e'*{'$$=$1*$2;}|e e '/'{'$$=$1/$2;}|num;
%%
int yylex()
{
    int c=getchar();
    if(isdigit(c))
    {
        yylval=c-'0';
        return num;
    }
    return c;
}
int yywrap()
{
    return 1;
}
int yyerror()
{
    return 1;
}
int main()
{
    yyparse();
    printf("\n");
    return 1;
}
```

OUTPUT:



```
imkiller@imkiller:~/CD/lab5$ yacc -v 2.y
imkiller@imkiller:~/CD/lab5$ gcc y.tab.c
imkiller@imkiller:~/CD/lab5$ ./a.out
34+
7
```

Q22. WAP in yacc to check strings of the following grammar:

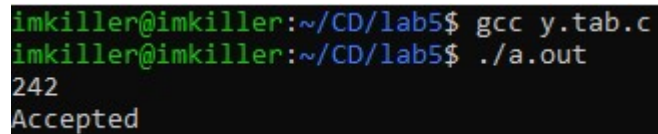
$S \rightarrow 2S2|3S3|4$

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

%%
E:S'\n'{printf("Accepted\n");}
S:'2'S'2'|'3'S'3'|'4';
%%

int yylex()
{
    return getchar();
}
int yywrap()
{
    return 1;
}
int yyerror()
{
    return 1;
}
int main()
{
    yyparse();
    printf("\n");
    return 1;
}
```

OUTPUT:



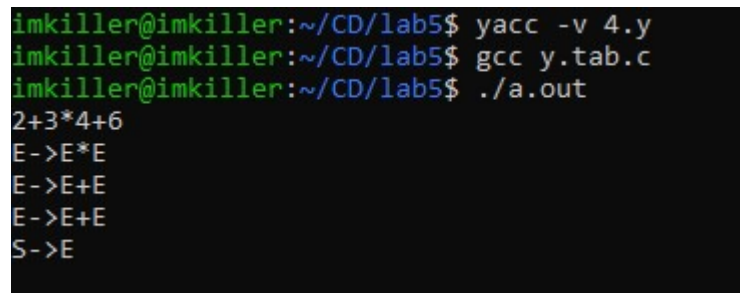
```
imkiller@imkiller:~/CD/lab5$ gcc y.tab.c
imkiller@imkiller:~/CD/lab5$ ./a.out
242
Accepted
```

Q23. WAP in yacc to check strings of the following grammar:

$E \rightarrow E + E \mid E * E \mid (E) \mid \text{Digit}$

```
%{
#include <stdio.h>
#include <ctype.h>
}%
%token DIGIT
%left '+'
%left '*'
%%
S:E'\n' {printf("S->E\n");}
E:E'+E' {printf("E->E+E\n");} | E'*E' {printf("E->E*E\n");} | ('E'){'$$=$2;} | DIGIT;
%%
int yylex()
{
    int c=getchar();
    if(isdigit(c))
    {
        yylval=c-'0';
        return DIGIT;
    }
    return c;
}
int yyerror()
{
    return 1;
}
int yywrap()
{
    return 1;
}
int main()
{
    yyparse();
    printf("\n");
    return 1;
}
```

OUTPUT:



```
imkiller@imkiller:~/CD/lab5$ yacc -v 4.y
imkiller@imkiller:~/CD/lab5$ gcc y.tab.c
imkiller@imkiller:~/CD/lab5$ ./a.out
2+3*4+6
E->E*E
E->E+E
E->E+E
S->E
```