System Software
Assignment 8

U20CS005
BANSI MARAKANA

1. Write a YACC and LEX program to implement a Calculator and recognize a valid arithmetic expression that uses operator +, -, *, /.

Lex program:

```
%{
        #include<stdio.h>
        #include "y.tab.h"
        extern int yylval;
%}
%%
[0-9]+ {yylval=atoi(yytext);return NUMBER;}
[\t] ;
[\n] return 0;
. return yytext[0];
%%
int yywrap()
{
        return 1;
}
```

Yacc program:

```
%{
   #include<stdio.h>
   int flag=0;
   void yyerror(const char *str)
   {
      printf("\nEntered arithmetic expression is Invalid");
      flag=1;
   }
%}
%token NUMBER
%left '+' '-'
%left '*' '/' '%'
%left '(' ')'
%%
ArithmeticExp: E{printf("Answer is: %d", $$);return 0;};
E:E'+'E {$$=$1+$3;}
|E'-'E {$$=$1-$3;}
|E'*'E {$$=$1*$3;}
|E'/'E {$$=$1/$3;}
|E'%'E {$$=$1%$3;}
|'('E')' {$$=$2;}
```

```
| NUMBER {$$=$1;};
%%
void main()
{
   printf("Enter an arithmetic expression: ");
   yyparse();
   if(!flag)
      printf("\nEntered arithmetic expression is Valid");
}
```

```
PS D:\BANSI MARAKANA\Yacc> ./a
Enter an arithmetic expression: 3+(4*5)-7
Answer is: 16
Entered arithmetic expression is Valid
```

2. Write a YACC and LEX program to check whether a given string is palindrome or not.

Lex program:

```
%{
        #include <stdio.h>
        #include "y.tab.h"
%}
%%
[a-zA-Z]+ {yylval.f = yytext; return String;}
[-+()*/] {return yytext[0];}
[\n]      {return 0;}
%%
int yywrap()
{
   return 1;
}
```

Yacc program:

```
%{
        #include <stdio.h>
        #include<stdlib.h>
        #include <string.h>
        extern int yylex();
        void yyerror(char *str);
        int flag, i, k = 0;
%}
%union {
        char* f;
}
%token <f> String
%type <f> E
%%
```

```
S : E {
        flag = 0;
        k = strlen($1) - 1;
        if(k%2==0)
        {
                for (i = 0; i <= k/2; i++)
                        if ($1[i] == $1[k-i]);
                        else
                                flag = 1;
                if (flag == 1)
                        printf("Entered string is not palindrome\n");
                else
                        printf("Entered string is palindrome\n");
                }
        else
        {
                for (i = 0; i < k/2; i++)
                        if ($1[i] == $1[k-i]);
                        else
                                flag = 1;
                if (flag == 1)
                        printf("Entered string is not palindrome\n");
                else
                        printf("Entered string is palindrome\n");
        }
};
E : String {$$ = $1;};
%%
void yyerror(char *str)
{
        fprintf(stderr, "%s\n", str);
        exit(1);
}
void main()
{
   printf("Enter a string: ");
   yyparse();
}
```

```
PS D:\BANSI MARAKANA\Yacc> ./a
Enter a string: ABCDCDCSA
Entered string is not palindrome
PS D:\BANSI MARAKANA\Yacc> ./a
Enter a string: abcdcdcba
Entered string is palindrome
```

3. Write a program for implementing given grammar for computing the expression using semantic rules of the YACC tool and LEX. Grammar: S-> SS* | SS+ | a

Lex program:

```
%{
   #include<stdio.h>
   #include "y.tab.h"
   extern int yylval;
%}
%%
a {yylval = 0;return A;}
[*] {yylval = 1;return B;}
[+] {yylval = 2;return C;}
.|\n {yylval = 3;return 0;}
%%
int yywrap()
{
   return 1;
}
```

Yacc program:

```
%{
   #include<stdio.h>
   #include<stdlib.h>
   int yyerror(char *msg)
   {
      printf("Invalid string\n");
      exit(0);
   }
%}
%token A B C
%%
R : S {printf("Valid string\n");};
S : S S B | S S C | A;
%%
void main()
{
   printf("Enter the string: ");
   yyparse();
}
```

```
PS D:\BANSI MARAKANA\Yacc> ./a
Enter the string: aaa*+
Valid string
PS D:\BANSI MARAKANA\Yacc> ./a
Enter the string: aaa
Invalid string
```

4. Write a YACC and LEX program to accept strings that start and end with 0 or 1.

Lex program:

```
%{
   extern int yylval;
   #include "y.tab.h"
%}
%%
0 {yylval = 0; return Zero;}
1 {yylval = 1; return One;}
.|\n {yylval = 2; return 0;}
%%
int yywrap()
{
        return 1;
}
```

Yacc program:

```
%{
   #include<stdio.h>
   #include<stdlib.h>
   void yyerror(const char *str)
   {
      printf("String Rejected!!\n");
      exit(0);
   }
%}
%token Zero One
%%
r : s {printf("String Accepted!!\n");};
s : n | One a | Zero b;
n : Zero | One;
a : n a | One;
b : n b | Zero;
%%
void main()
{
        printf("Enter a string to check whether it starts and ends with 0/1 or not: ");
        yyparse();
}
```

```
PS D:\BANSI MARAKANA\Yacc> ./a
Enter a string to check whether it ends with 0/1 or not: 1001
String Accepted!!
PS D:\BANSI MARAKANA\Yacc> ./a
Enter a string to check whether it ends with 0/1 or not: 100
String Rejected!!
```