NAME: VAIBHAV BANKA

REG NO: 21BCE1955

EXPERIMENT NUMBER: 8

SUBJECT: COMPILER DESIGN

1. Write a lex program to convert decimal to binary.

```
%{
 #include<stdio.h>
 #include<math.h>
 void bin(int x);
 int x;
%}
%%
[0-7]+ {x=atoi(yytext);
        bin(x);
        return x;}
%%
int main(){
 yylex();
 return 0;
}
void bin(int x){
 int b=0,i=0;
 while(x>1){
  int r=x%2;
  int res=pow(10,i);
  b+=r*res;
  x=x/2;
  i++;
 }
 b+=x*pow(10,i);
 printf("Binary number is: %d\n",b);
}

int yywrap(){}
```

```
%{
  #include<stdio.h>
  #include<math.h>
  void bin(int x);
  int x;
%}
%%
[0-7]+ {x=atoi(yytext);
        bin(x);
        return x;}
%%
int main(){
  yylex();
  return 0;
}
void bin(int x){
  int b=0,i=0;
  while(x>1){
    int r=x%2;
    int res=pow(10,i);
    b+=r*res;
    x=x/2;
    i++;
  }
  b+=x*pow(10,i);
  printf("Binary number is: %d\n",b);
}

int yywrap(){}
```

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ lex ex8q1.l
vaibhav@vaibhav-virtual-machine:~$ cc lex.yy.c -lm
vaibhav@vaibhav-virtual-machine:~$ ./a.out
10
Binary number is: 1010
vaibhav@vaibhav-virtual-machine:~$ ./a.out
12
Binary number is: 1100
vaibhav@vaibhav-virtual-machine:~$ ./a.out
36
Binary number is: 100100
vaibhav@vaibhav-virtual-machine:~$
```

2. Write a yacc program to convert decimal to binary

LEX

```
%{
 #include"y.tab.h"
 extern int yylval;
%}
%%
[0-7]+ {yylval=atoi(yytext);
          return num;}
%%


int yywrap(){}
```

```
%{
  #include"y.tab.h"
  extern int yylval;
%}
%%
[0-7]+ {yylval=atoi(yytext);
         return num;}
%%

int yywrap(){}

```

YACC

```
%{
 #include<stdio.h>
 #include<math.h>
 int yylex(void);
 int yyerror(char *s);
 void bin(int);
%}
%token num
%%
S:num{bin($1); return 0;};
%%


void bin(int x){
 int b=0,i=0;
 while(x>1){
   int r=x%2;
   int res=pow(10,i);
```

```
%{
  #include<stdio.h>
  #include<math.h>
  int yylex(void);
  int yyerror(char *s);
  void bin(int);
%}
%token num
%%
S:num{bin($1); return 0;};
%%

void bin(int x){
  int b=0,i=0;
  while(x>1){
    int r=x%2;
    int res=pow(10,i);
    b+=r*res;
    x=x/2;
    i++;
  }
  b+=x*pow(10,i);
  printf("Binary number is: %d\n",b);
}
int main(){
  yyparse();
  return 0;
}

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

```
    b+=r*res;
    x=x/2;
    i++;
  }
  b+=x*pow(10,i);
  printf("Binary number is: %d\n",b);
}
int main(){
  yyparse();
  return 0;
}

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

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ lex ex8q2.l
vaibhav@vaibhav-virtual-machine:~$ yacc -d ex8q2.y
ex8q2.y:11 parser name defined to default :"parse"
vaibhav@vaibhav-virtual-machine:~$ cc lex.yy.c y.tab.c -lm
vaibhav@vaibhav-virtual-machine:~$ ./a.out
15
Binary number is: 1111
vaibhav@vaibhav-virtual-machine:~$ ./a.out
32
Binary number is: 100000
vaibhav@vaibhav-virtual-machine:~$ ./a.out
46
Binary number is: 101110
vaibhav@vaibhav-virtual-machine:~$
```

3. Write a lex program to convert binary to decimal.

```
%{
 #include<stdio.h>
 #include<math.h>
 void bin(int x);
 int x;
%}
%%
[0-1]+ {x=atoi(yytext);
        bin(x);
        return x;}
. {printf("Invalid String");};
%%
int main(){
 yylex();
 return 0;
}
void bin(int x){
 int b=0,i=0;
 while(x>1){
  int r=x%10;
  int res=pow(2,i);
  b+=r*res;
  x=x/10;
  i++;
 }
 b+=x*pow(2,i);
 printf("Decimal number is: %d\n",b);
}

int yywrap(){}
```

```
%{
  #include<stdio.h>
  #include<math.h>
  void bin(int x);
  int x;
%}
%%
[0-1]+ {x=atoi(yytext);
        bin(x);
        return x;}
. {printf("Invalid String");};
%%
int main(){
  yylex();
  return 0;
}
void bin(int x){
  int b=0,i=0;
  while(x>1){
    int r=x%10;
    int res=pow(2,i);
    b+=r*res;
    x=x/10;
    i++;
  }
  b+=x*pow(2,i);
  printf("Decimal number is: %d\n",b);
}

int yywrap(){}
```

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ lex ex8q3.l
vaibhav@vaibhav-virtual-machine:~$ gcc lex.yy.c -lm
vaibhav@vaibhav-virtual-machine:~$ ./a.out
1010
Decimal number is: 10
vaibhav@vaibhav-virtual-machine:~$ ./a.out
10000
Decimal number is: 16
```

4. Write a yacc program to convert binary to decimal.

LEX

```
%{
 #include"y.tab.h"
 extern int yylval;
%}
%%
[0-7]+ {yylval=atoi(yytext);
        return num;}
%%
int yywrap(){}
```

```
%{
  #include"y.tab.h"
  extern int yylval;
%}
%%
[0-7]+ {yylval=atoi(yytext);
        return num;}
%%

int yywrap(){}
```

YACC

```
%{
 #include<stdio.h>
 #include<math.h>
 int yylex(void);
 int yyerror(char *s);
 void bin(int);
%}
%token num
%%
S:num{bin($1); return 0;};
%%
void bin(int x){
 int b=0,i=0;
 while(x>1){
  int r=x%10;
  int res=pow(2,i);
  b+=r*res;
  x=x/10;
```

```
%{
  #include<stdio.h>
  #include<math.h>
  int yylex(void);
  int yyerror(char *s);
  void bin(int);
%}
%token num
%%
S:num{bin($1); return 0;};
%%

void bin(int x){
  int b=0,i=0;
  while(x>1){
    int r=x%10;
    int res=pow(2,i);
    b+=r*res;
    x=x/10;
    i++;
  }
  b+=x*pow(2,i);
  printf("Decimal number is: %d\n",b);
}

int main(){
  yyparse();
  return 0;
}

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

```
    i++;
  }
  b+=x*pow(2,i);
  printf("Decimal number is: %d\n",b);
}
int main(){
  yyparse();
  return 0;
}
int yyerror(char *msg){
  printf("invalid string\n");
}
```

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ lex ex8q4.l
vaibhav@vaibhav-virtual-machine:~$ yacc -d ex8q4.y
ex8q4.y:11 parser name defined to default :"parse"
vaibhav@vaibhav-virtual-machine:~$ cc lex.yy.c y.tab.c -lm
vaibhav@vaibhav-virtual-machine:~$ ./a.out
11110
Decimal number is: 30
vaibhav@vaibhav-virtual-machine:~$ ./a.out
100100
Decimal number is: 36
vaibhav@vaibhav-virtual-machine:~$
```

5. Write a yacc program to convert infix to postfix

LEX

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

int yywrap(){}
```

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

int yywrap(){}
```

YACC

```
%{
#include<stdio.h>
int yyerror (char *s);
int yylex(void);
%}
%token NUM
%left '+' '-'
%left '*' '/'
%right NEGATIVE
%%
S:  E {printf("\n");}
   ;
E:  E '+' E {printf("+");}
   |  E '*' E {printf("*");}
   |  E '-' E {printf("-");}
   |  E '/' E {printf("/");}
   |  '(' E ')'
```

```
%{
#include<stdio.h>
int yyerror (char *s);
int yylex(void);
%}
%token NUM
%left '+' '-'
%left '*' '/'
%right NEGATIVE
%%
S:  E {printf("\n");}
    ;
E:  E '+' E {printf("+");}
    |   E '*' E {printf("*");}
    |   E '-' E {printf("-");}
    |   E '/' E {printf("/");}
    |   '(' E ')'
    |   '-' E %prec NEGATIVE {printf("-");}
    |   NUM     {printf("%d", yylval);}
    ;
%%

int main(){
 printf("Enter the infix expression\n");
 yyparse();
}

int yyerror (char *s) {
    printf("\n Expression is invalid\n");
}
```

```
    |   '-' E %prec NEGATIVE {printf("-");}
    |   NUM     {printf("%d", yylval);}
    ;
%%

int main(){
 printf("Enter the infix expression\n");
 yyparse();
}

int yyerror (char *s) {
    printf("\n Expression is invalid\n");
}
```

OUTPUT

```
vaibhav@vaibhav-virtual-machine:~$ lex ex8q5.l
vaibhav@vaibhav-virtual-machine:~$ yacc -d ex8q5.y
ex8q5.y:21 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 infix expression
1+2*3+4
123*+4+
vaibhav@vaibhav-virtual-machine:~$ ./a.out
Enter the infix expression
2-3+4-5*6
23-4+56*-
vaibhav@vaibhav-virtual-machine:~$ ./a.out
Enter the infix expression
(2-3+4)*(5+6*7)
23-4+567*+*
vaibhav@vaibhav-virtual-machine:~$
```