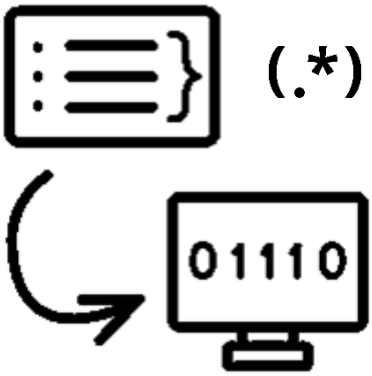
Atma Ram Sanatan Dharma College
University of Delhi



System Programming – YACC

Practical File for Paper Code 32347501

Submitted By
Sudipto Ghosh
College Roll No. 19/78003
BSc (Hons) Computer Science

Submitted To
Ms Manisha Bagri
Department of Computer Science

INDEX

S No.	Objective	Date	Sign
1	A Program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).	--	
2	Program in YACC to recognize the language ($a^n b$, n ≥ 10) (Output to say input is valid or not)	--	
3	A Program in YACC which recognizes a valid variable which starts with letter followed by a digit. The letter should be in lowercase only.	--	
4	Program in YACC to recognize the strings "ab", "aabb", "aaabbb", … of the language ($a^n\, b^n$, n ≥ 1).	--	

PRACTICAL 1

Objective

A Program in YACC to evaluate an expression (simple calculator program for addition and subtraction, multiplication, division).

Code

```
/* yacc.y */
%{
  #include <stdio.h>
  #include <stdlib.h>
  extern int yylex();
  void yyerror(char *);
%}

%union { float f; }
%token <f> NUM
%type <f> E T F
%%
S : E { printf("%f\n", $1); }
  ;
E : E '+' T { $$ = $1 + $3; }
  | E '-' T { $$ = $1 - $3; }
  | T
  ;
T : T '*' F { $$ = $1 * $3; }
  | T '/' F { $$ = $1 / $3; }
  | F
  ;
F : '(' E ')' { $$ = $2; }
  | '-' F { $$  = -$2; }
  | NUM
  ;
%%
int main()
{
   yyparse();
   return 0;
}

void yyerror(char *msg) {
  fprintf(stderr, "%s\n", msg);
  exit(1);
}

/* lex.l */
%{
  #include <stdio.h>
```

```
  #include <stdlib.h>

  #if __has_include("y.tab.h")
    #include "y.tab.h"
  #endif
%}

%option noyywrap
%%
[0-9]+(\.[0-9]+)? { yylval.f = atof(yytext); return NUM; }
[\-+()*/] { return yytext[0]; }
[ \t\n]+ { ; }
%%
```

Output

```
$ .\a
5+5
10.000000
$ .\a
2*10
20.000000
$ .\a
50/2
25.000000
$ .\a
2-6
-4.000000
```

PRACTICAL 2

Objective

Program in YACC to recognize the language ($a^n b$, n ≥ 10) (Output to say input is valid or not)

Code

```
/* yacc.y */
%{
  #include <stdio.h>
  #include <stdlib.h>
  extern int yylex();
  void yyerror(char *);
%}

%token A B
%%
S : X Y B '\n' { printf("VALID STRING\n"); exit(0); }
  ;
X : A A A A A A A A A A
  ;
Y : A Y
  |
  ;
%%
int main()
{
   yyparse();
   return 0;
}

void yyerror(char *msg) {
  fprintf(stderr, "INVALID STRING\n");
  exit(1);
}

/* lex.l */
%{
  #include <stdio.h>
  #include <stdlib.h>

  #if __has_include("y.tab.h")
    #include "y.tab.h"
  #endif
%}

%option noyywrap
%%
[a] { return A; }
```

```
[b] { return B; }
[ |\n|\t] { return yytext[0]; }
. { return yytext[0]; }
%%
```

Output

```
$ .\a
a
INVALID STRING
$ .\a
a
INVALID STRING
$ .\a
aaaaaaaaab
VALID STRING
$ .\a
aaaaaaaaabb
INVALID STRING
$ .\a
aaaaaaaaaab
VALID STRING
$ .\a
ab
INVALID STRING
```

PRACTICAL 3

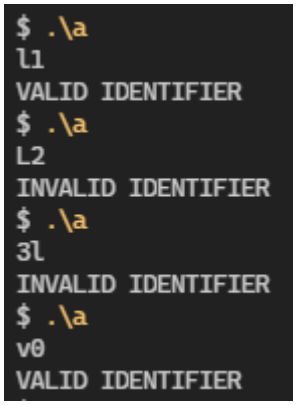
Objective

A Program in YACC which recognizes a valid variable which starts with letter followed by a digit. The letter should be in lowercase only.

Code

```
/* yacc.y */
%{
  #include <stdio.h>
  #include <stdlib.h>
  extern int yylex();
  void yyerror(char *);
%}

%token D L
%%
S : L D { printf("VALID IDENTIFIER\n"); }
  ;
%%
int main()
{
    yyparse();
    return 0;
}

void yyerror(char *msg)
{
    fprintf(stderr, "INVALID IDENTIFIER\n");
    exit(1);
}

/* lex.l */
%{
  #include <stdio.h>
  #include <stdlib.h>

  #if __has_include("y.tab.h")
    #include "y.tab.h"
  #endif
%}
%option noyywrap
%%
[a-z] { return L; }
[0-9] { return D; }
[ \t\n]+ { ; }
. { return yytext[0]; }
%%
```

Output


$.\a
l1
VALID IDENTIFIER
$.\a
L2
INVALID IDENTIFIER
$.\a
3l
INVALID IDENTIFIER
$.\a
v0
VALID IDENTIFIER

PRACTICAL 4

Objective

Program in YACC to recognize the strings "ab", "aabb", "aaabbb", … of the language ($a^n b^n$, n ≥ 1).

Code

```
/* yacc.y */
%{
  #include <stdio.h>
  #include <stdlib.h>
  extern int yylex();
  void yyerror(char *);
%}

%token A B

%%
S : E '\n' { printf("VALID STRING\n"); exit(0); }
  ;
E : A E B
  | A B
  ;
%%
int main()
{
   yyparse();
   return 0;
}

void yyerror(char *msg) {
  fprintf(stderr, "INVALID STRING\n");
  exit(1);
}

/* lex.l */
%{
  #include <stdio.h>
  #include <stdlib.h>

  #if __has_include("y.tab.h")
    #include "y.tab.h"
  #endif
%}

%option noyywrap
%%
[a] { return A; }
[b] { return B; }
```

```
[ |\n|\t] { return yytext[0]; }
. { return yytext[0]; }
%%
```

Output