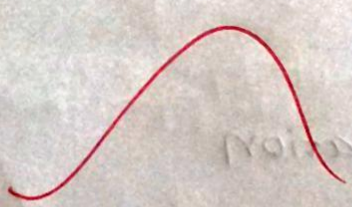
EXPERIMENT-1

P. VISHNU SAI

AIM:- To implement a C program to perform Symbol Table operations.

PROGRAM

```
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#define NULL 0
int size=0
void insert();
void Display();
void Delete();
Case:1
Insert();
break;
Case:-2:
Display();
break
Case;3
Delete();
break;
Case;4
printf("\n\t Enter the label to be searched");
scanf("%s",la);
y=a search(la);
if(y==1)
printf("\n\t The label is present in the Symbol
                                table \n")
```



```
Else
printf("\n\t The Label is not present in the Symbol
        table\n");
break;
int a;
char[10];
struct SymbTab *p,*q;
p=first;
printf("\n Enter the Lab to be deleted :");
Scanf("%s",l);
a=Search(l);If(a==0)
Printf("\n\t Label not found\n").
Size--;
Printf("\n\t After Deletion :\n");
Display()
 }
 }
```

Sample Output

Symbol table implementation

INSERT

DISPLAY

DELETE

SEARCH

MODIFY

EXIT

Result: C program for Symbol pattern representation is Executed.

EXPERIMENT-2

P. VISHNU SAI

AIM: To write c program for identify identifiers, Constants and operations.

Program :-

```
#include<string.h>
#include<ctype.h>
#include<stdio.h>
FILE *f1,*f2,*f3;
char c,str[10],st[10];
int num[100],lineno=0,tokenvalue=0,i=0,j=0,
  k=0;
f1=fopen("input","w");
while((c=getchar())!=EOF) put(c,f1);
fclose(f1);
f1=fopen("input","r");
f2=fopen("identifier","w");
f3=fopen("specialchar","w");
while(isdigit(c)||isalpha(c)||c=='_'||c==$)
{
 put(c,f2);
 c=getc(f1);
}
put(' ',f1)
ungetc(c,f1);
}
```

```
if(c=='+'||c=='-')
printf(" ");
else if(c=='\n')lineno++;
else
putc(c,fs);
}
fclose(fs);
fclose(fk);
fclose(fi);
printf("\n The no's in the program are:");
for(j=0;j<i;j++)
printf("%d,num[j]);
printf("\n");
fk=fopen("identifier","r");
if(c!='') st1[k++]=c;
else
st1[k]='\0';
keyword(st1);
k=0;
}
}
fclose(fs);
```

Output:- variables Operator : -+ : 2 constants

2136599 -25

Keywords : int if then Else End if special Symbol : ()

Comments : this a comment line

Result:- C program to identify, constants & Operators.

Experiement-3

P. VISHNU 501

Aim:- Implement a C program to eliminate left recursion from a given CFG.

program

```
#include <stdio.h>
#include <string.h>
#define SIZE 10
int main(){
char non-terminal;
char beta, alpha;
int num;
char production[10][SIZE];
int index=3;
printf("Enter no of production :");
scanf("%d", &num);
printf("Enter the grammer E->E->A :\n");
for(int i=0; i<n; i++)
{
scanf("%s", production[i]);
}
non-terminal = production[i][index]){
alpha = production[i][index+1]
printf("is left recursive :\n");
while(production[i][index] != 0)
printf("%c->%c%c|", non-terminal, beta,
                    non-terminal);
```

```
printf("\n %c -> %c %c",non-terminal,
alpha, non-terminal);
}
else
printf("can't be reached);
}
else
printf("is not left recursive) inder -3
}
}
```

Sample Output

Enter Number of production: 4 Enter the grammer E→EA : E→EA|A

A→AT|aT→a

E→i

GRAMMER ::: T→a is not left recursive

GRAMMER ::: E→ is not left recursive

Result

To implement a C program to Eliminate left recursion CFG is Executed Successfully.

4. Left factoring

P VISHNU 501

AIM:- To implement a c program to Eliminate left factoring from a given CFG.

program:-

```
#include <stdio.h>
#include <string.h>
int main()
{
 char gram[20], part1[20], part2[20], modified[20];
 int i, j=0, k=0, j=0, pos;
 Printf("Enter production : A->");
 gets(gram);
 part1[j]='\0';
for(i=0; i<strlen(part1)||i<strlen(part2); i++)
 {
  if(part1[i]==part2[i])
 {
  modifiedGram[k]=part1[i]; k++;
   pos=i+1;
  }
  }
  newGrand[j+1]='|';
for(i=pos; j=0, part1[j]; j++).
```

3/8/17

Output

```
Enter production A->abc|abcFG
    A->abcX  X->DEFG
```

Result:- c program for left factoring is Executed.

5. COUNT THE NUMBER OF CHARACTERS, WORDS, LINES

AIM: To write a C program for implementing a lexical Analyzer to scan and count the number of characters, words & lines in a file.

PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
int main(){
    FILE *file;
    char path[100];
    char ch;
    int characters, words, lines;
    printf("Enter source file path");
    scanf("%s", path);
    file = fopen(path, "r");
    if (file == NULL)
    printf("\n Unable to open file\n");
    printf(" please check if file exists & you
        have read privilege);
    Exit (EXIT_FAILURE);
}
characters = words = lines = 0;
while (ch = fgetc(file) != EOF){
        characters;
```

```
if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
    words++;
}
if (characters > 0) {
    words++;
    lines++;
}
printf("\n");
printf("Total characters = %d\n", characters);
printf("Total words = %d\n", words);
printf("Total lines = %d\n", lines);
return 0;
}
```

Output

```
Enter source file path: input.txt
Total characters: 8
Total words: 2
Total lines: 1
```

Result:- c program for implementing a lexical Analysis to scan & count the no of characters, words & lines in a file.

6. COMPUTATION OF FIRST

AIM:- To write a C program for to find first & follow for the predictive parser.

PROGRAM

```
#include<stdio.h>
#include<ctype.h>
void FIRST(char[],char);
void addToResultSet(char[],char);
int numOfProductions;
char productionSet[10][10];
int main()
{
 int i;
 char choice;
 char c
 char result[20];
 printf("How many no of production");
 scanf("%s",production set[i]);
  }
do
{
printf("\n Find the FIRST of:");
 scanf("%c",&choice);
 }
 while(choice=='y'||choice=='y')
  }
```

```
void FIRST (char* Result, char c)
{
int i,j,k;
char subResult[20];
int foundEpsilon;
subResult[0]='!0';
Result[0]='!0';
return;
}
int k;
for(k=0; Result[k]!='!0'; k++)
if(Result[k]==val)
return;
Result[k]=val;
Result[k+1]='!0';
}
```

Output :-

Enter productions No. : E=TD

Enter production No. D=+TD

Enter " " D=$

" " T=FS

" " S=*FS

" " S=$

Result :- C program for Computation 0-1 Test is Executed Successfully.

7 Computation of FOLLOW

AIM:- Write a C program to find FOLLOW Set for predictive parser.

Program :-

```
#include <stdio.h>
#include <ctype.h>
char production[10][10], array[10];
int main()
{
int count;
char option, ch;
printf("Enter Total no of productions:");
scanf("%d", &limit);
for (count=0; count<limit; count++)
 }
do
x=0;
printf("Enter productor value to find follow");
scanf("%c", &ch);
void find_follow(char c)
 {
  int i,j;
  int length = strlen(production[i]);
   if(production[0][0]==ch)
```

```
void find_first (char ch)
{
int i, k; if (! is Upper (ch))
{
Array - Manipulation (ch)
 }
if ( production [i] [0] = -$)
{
find -follow (production [i] [0]);
}
Else if (!is lower (production [k] [2]);
 }
Else
{
 find - first (production [k] [2]);
}
}
}
```

Output:

Enter total no of productions 8

D

Follow value of D

To continue, press Y, n:

3/8/25

Result: C program to find follow is successfully completed.

8 COMMENT OR NOT

AIM:- To implement c program to identify whether a given line is comment or not.

Program:-

```
#include <stdio.h>
#include <conio.h>
Void main()
{
char com[30];
int i=0, a=0;
Printf(" Enter comment:");
a=1
break;
if (a==0)
Printf(" It is not a comment);
Else {
printf (" It is comment);
}
```

Output:-

```
Enter comment: hello
It is Comment.
```

Result:- C program to identify whether a given line is Comment or not is Completed.

9: Identifier or not.

P. VISHNU 501

Aim:- Write a cprogram to test whether given identifier is valid or not.

Program:-

```
#include <stdio.h>
#include<conio.h>
#include <ctype.h>
Void main()
{
char a[10];
int flag, i=1;
clrscr();
Printf(" Enter a identifier:");
gets(a);
if()isalpha(a[0])
flag=1; else
Printf ("not valid");
while (a[i]!='0')
  {flag=0 ; break
  }
  getch();
}
```

Output :- First valid identifier.

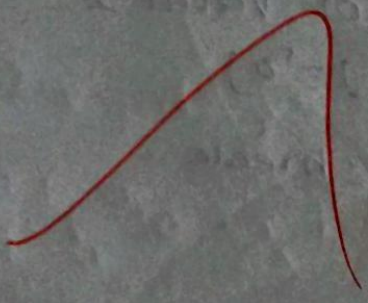
Result:- The cprogram for identifier is successfully completed.

10. Predictive Parsing

AIM:- To write a c program for Constructing of LL(1) parsing.

Program:-

```
#include <stdio.h>
#include <conio.h>
int main()
{
char m[5][6][3] = {"tb. ":" ": };
int size[5][6]
int j, k, n, str1, str2;
Printf(" Enter the Input String ");
Scanf("%s", &s)
Switch (stack[n])
{
case 'e';
str1 = 'b';
break;
Case 'b';
str1 = 1
break;
Case t
str1 = 2;
break;
case 't';
str1 = 3;
break;
```



P. Vishnu 501

```
Switch (str[i])
{
case 'i':
str2 = 0;
 break;
case '+':
str2 = 1;
break;
case '*':
str2 = 2;
 break;
case '(':
 str1 = 3;
 break;
 case ')':
  str2 = 4;
   break;
   case $:
   str2 = 5;
    break;
}
Printf("In Success").
   getch();
 }
```

Output:-

```
$bt  i*i+i
   $b & SUCCESS
```

Result:- C program for constructing LL is completed.

4. Computation of First

Aim:- To Execute c program to construct recursive descent parsing.

Program

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
char input[10];
int i,l;
void main();
if (input[i+1]=='0')
  printf("\n string is accepted");
else
 printf("\n string is not accepted");
getch();
else
return(0);
EP()
{
if (input[i]=='+')
   {
   i++;if T()
 {
   if (EP())
   return(1);
   else
```

```
return 0;
Else
if E()
 return(1)
 }
if(input[i]=='$')
 {
 i++;
 if E(1)
 Else
 return(0);
 }
 Else
 return(0);
 }
 return(0);
```

Output:-

$E \rightarrow TE'$

$E \rightarrow +TE' | @$

$T \rightarrow FT'$

$T' \rightarrow *FT' | @$

$F \rightarrow (E) | ID$

Result:-

The c program to construct recursive descent parsing is executed successfully.

12. STACK- SHIFT REDUCE PARSER

Aim:- To write a c program for stack implementation to the shift Reduce power.

Program

```
#include<stdio.h>
#include<stdlib.h>
char ip-sym[], stack[5]
char act[15];
int main():{
Printf("SHIFT REDUCE PARSER");
Printf("\n GRAMMER");
len = strlen(ip_sym);
 ip-sym[ip-ptr] = '';
  ip-ptr=;
 str-ptr++
  }
 str - ptr++;
  check();
  }
if (flag == 0)
{ Exit(0);
return
1.
```

o/p a+b= $ a+b$ --

$a +b$ shift a

Result:- C program for stack-shift reduce parser is executed.

13. Operator precedding parser.

P. Vishnu 99

Aim:- To write a c program for Operator precedence parsing.

Program

```
#include<stdio.h>
#include<string.h>
char input[20];
char lasthandle[6],stack[50],handles[][5]
top=0,i;
char pre[9][9]={
Switch(){
case '+':
 return 0;
Case '-'
  return 1;
case 'a'
return 2
case '.'
  return 3;
 }
return 0;
 }
```

Op

$i →(i+1)*p Shift

$R →(i+1)*s Reduced

Result:- c program for Operation precedence parsing Executed Successfully.