5. no. of characters/line

P. Vinnu Vardhan Reddy
Reg no:- 192011179

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 <stdio.h>
int main(){
file *file;
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 EXIT")
printf("Please check if file EXIS & you have read
exit(FNT failure)
}
characters = words = lines = 0
while(ch = fgetc file = EOF)
    characters;
if(ch == ' ' || ch = (+ch == '\n' || ch == 10)
words ++;
}
```

P. Vishnu Vardhan Reddy
Reg no: 192011179.

```
if (characters > 0) {
words ++;
lines ++;
}
printf ("\n")
Printf ("\nTotal character = %d\n", characters);
Printf ("\nTotal words = %d\n", words);
Printf ("\nTotal lines = %d\n", lines);
return 0;
}
```

output:-

Enter source file path input.text.

Total character = 8.

Total words = 2.

Total lines = 1.

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

6. Computation of first: 192011179.

Aim: To write a C program for a find first & follow for the predictive parser.

Program:

```
#include<stdio.h>
#include<type.h>
void first(char,char);
void add to result set(char*,char);
if num of productions;
char production set T(10)[10];
int main()
{
int i
char choice;
char c;
char result[20];
printf("How many no. of production");
scanf("%d production set(i);
}
do
}
printf("\n find the first of:");
scanf("%d",& choice);
}
```

```
while(choice==1y1||choice.c==1y1)
 }.
void first(char* Result[20];
int found Epsilion;
sub Result[0]=1!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';
  }.
```

P. vimmu
192011749.

output:-

Enter production m1 = E=TD.
Enter production m2 = D=*TD.
Enter production m3 = T=$
Enter production m4 = $=F*
Enter production m5 = $=$.

Result:- c program for computation of first
is Executed succesfully.

7. Computation of follow

Aim :- Write a c program to find follow set for Predictive parser.

Program :-

```
#include<stdio.h>
#include<type.h>
char production[10][10],array[10];
int main()
{
int count;
char option,ch;
Printf("Enter total no of production");
Scanf("%d",&unit");
for(count=0;count<limit;count++);
}
do
x=0;
Printf("Enter production value to find follow")
Scanf("%c",&ch);
void find follow (char)
{
limit 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 = match population (ch)
}
if (production [k] [0] == '#')
{
find follow (production [i] [0]);
}
else if (is lower (production [k] [2]));
}
else
{
find - first (production [k] [2]) ;
}
}
}

output:
Enter total no. of production
D
follow value of D:
To continue press

Result: program to find follow is successfully completed.

8. COMMENT OR NOT

P. Vishnu 160311729 20

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=2, a=0;
Printf("Enter comment");
 a=1
break;
 if (a==0)
Printf("It is not a comment");
}
else {
 Print f("It is comment");
 3.
```

Output:

Enter comment hello

It is comment.

Result: c Program to identify whether a given line is comment a not is completed

9. Identifier or not.

P. Vishnu 2/
192011179.

Aim: write a c program to test whether given identifier is valid or not.

Program:

```
#include<stdio.h>
#include<conio.h>
#include<type.h>
void main()
{
char a[10];
int flag; i=1;
clsr();
printf("Enter a identifier:)
gets(a);
printf is alpha(a(0)).
flag=1; else.
printf("Not valid");
while(a[i]!='0').
{ flag=0; break.
} getch();
}
```

output: Ant.
valid identifier.

Result: The c program for identifier successfully completed.

10. Predictive Parsing

Putmm. 22
19201094.

Aim: To write a C program for constructing of (LL1) parsing.

Program:

```
#include <stdio.h>.
#include <conio.h>.
int main().
{.
char m[5][6][3] = {"/b, " ", " ", };
int size (5) (6)
int i, k, n, str, str2;
printf ("Enter the input string");
scanf ("%s", &x);
switch (stack(1));
{
case 'e',
str1='0';
break;
case 'b':
str1=1
break;
case E.
str1=2
break;
case 'c':
str1=3;
break;
```

```
switch(str1);
{
case'i';
str2=0';
  break;
case'+';
Str2=1;
case'K';
str2=2;
break;
case'c';
str=3;
break;
case';'
str2=4;
break;
case'8';
str2=5;
  break;
}.
print("\n success");
getch();
}.
```

output;
8bt i&js.
8b8 success

Result:- C program for constructing LL is completed.

23/8/23

P. Vishnu.
1930111A9.

Aim:- TO Execute a program to construct recursive descent parsing.

Program:-

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
char input[100];
int i,l;
void main();
if(input[i+1]=='0')
Printf("String is accepted");
else
printf("String is not accepted");
getch();
else
return(0);
EP()
{
if(input[i]=='+')
{
i++; if T()
{
if(EP())
return(1);
else
```

P. Vijitha
192011189

```
return(0);
else
if(E())
return(1)
}
if(input[i]==')')
{
i++;
if E(cb)
else.
return(0);
}
else
return(0);
}
return(0);
```

output:

E → TE'
E → TE'||a
T → FT'
T' → *FT'|ε
F → (E)||D.

Result:
The C program to construct recursive descent parsing is executed successfully.

12. STACK-Shift Reduce Parser

Aim:. To write a program for stack implementation for the shift Reduce Parser

Program:

```
#include <stdio.h>
#include <stdlib.h>
char ip. sym[5], stack[5]
charact[15];
int main(); {
Printf("shift Reduce PARSER");
Printf("\n Grammer");
len=strlen(ip. sym);
ip-sym[ip-ptr]='';
ip-ptr=;
str.pt++
}
str-ptr++;
check();
}
if flag==0)
Exit(0);
}
return.
}
```

o/p ath = x athb - -
(a thb. shift.

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

13. Operation Precedding Parser. Prithvi 22
1920111749

Aim:- To write a C Program for operation Precedure Parsing.

Program:-

```
#include <stdio.h>
#include <string.h>
char input int=0;
char[outhandle, stack[50], handle[3][5]
top=0,1;
char Pre[9][9]={
switch(c))
{
case '+';
return 0;
case :-'
return*
case'*'
return 2
case'('
return 3';
}
return0;
}
```

O/P:-

~ $i & (i+1)$ as shift.
$R & (i+1)$ as Reduced.

Result:-
C program for operation precedure passing executed successfully.