Write a C program to implement Recursive Descent parser for the following grammar

S → (L)/a

L → SL’

L’ → ,SL’/∈

PROGRAM :

#include <stdio.h>

#include <string.h>

int S();

int Ldash();

int L();

char \*ip;

char string[50];

int main()

{

printf("Enter the string : ");

scanf("%s",&string);

ip = string;

printf("\n-------------------------------------------------------------");

printf("\n\nInput\t\tAction\n\n");

if (S())

{

printf("\n-------------------------------------------------------------");

printf("\nString is Successfully Parsed\n\n\n");

}

else

{

printf("\n-------------------------------------------------------------");

printf("\nError in parsing string\n");

}

}

int L()

{

printf("%s\t\tL-->SL' \n",ip);

if (S())

{

if(Ldash())

{

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

int Ldash()

{

if(\*ip ==',')

{

printf("%s\t\tL-->,SL' \n",ip);

ip++;

if(S())

{

if (Ldash())

{

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

else

{

printf("%s\t\tL'-->e \n",ip);

return 1;

}

}

int S()

{

if(\*ip =='(')

{

printf("%s\t\tS-->(L) \n",ip);

ip++;

if(L())

{

if (\*ip ==')')

{

ip++;

return 1;

}

else

{

return 0;

}

}

else

{

return 0;

}

}

else if (\*ip == 'a')

{

ip++;

printf("%s\t\tS-->a \n",ip);

return 1;

}

else

{

return 0;

}

}

OUTPUT :

Recursive descent parsing for the following grammar

E->TE'

E'->+TE'/@

T->FT'

T'->\*FT'/@

F->(E)/ID

Enter the string to be checked:(a+b)\*c

String is accepted

Recursive descent parsing for the following grammar

E->TE'

E'->+TE'/@

T->FT'

T'->\*FT'/@

F->(E)/ID