WEEK 6

Nikitha Godavarthi
AP19110010460
CSE-C

Problem 2:

Construct Recursive Descent Parser for the grammar

G = ({S, L}, {(,), a, ,}, {S → (L) | a ; L → L, S | S}, S) and verify the acceptability of the following strings:

i. (a,(a,a))
ii. (a,((a,a),(a,a)))

You can manually eliminate Left Recursion if any in the grammar.

CODE:

```
#include<stdio.h>
#include<string.h>
int S(),Ldash(),L();
char *ip;
char string[50];
int main()
{
   printf("Enter the string\n");
   scanf("%s",string);
   ip=string;
   printf("\n\nInput\t\tAction\n");
   if(S() && *ip=='\0')
   {
      printf("\n String is successfully parsed\n");
   }
   else
   {
      printf("Error in parsing String\n");
   }
}
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;
   }
}
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'->^ \n",ip);
     return 1;
  }
}
```

OUTPUT:

```
Enter the string
(a,(a,a))


Input           Action
(a,(a,a))               S->(L)
a,(a,a))                L->SL'
,(a,a))         S->a
,(a,a))         L'->,SL'
(a,a))          S->(L)
a,a))           L->SL'
,a))            S->a
,a))            L'->,SL'
))              S->a
))              L'->^
)               L'->^

 String is successfully parsed


...Program finished with exit code 0
Press ENTER to exit console.
```

```
Input           Action
(a,((a,a),(a,a)))               S->(L)
a,((a,a),(a,a)))                L->SL'
,((a,a),(a,a)))         S->a
,((a,a),(a,a)))         L'->,SL'
((a,a),(a,a)))          S->(L)
(a,a),(a,a)))           L->SL'
(a,a),(a,a)))           S->(L)
a,a),(a,a)))            L->SL'
,a),(a,a)))             S->a
,a),(a,a)))             L'->,SL'
),(a,a)))               S->a
),(a,a)))               L'->^
,(a,a)))                L'->,SL'
(a,a)))         S->(L)
a,a)))          L->SL'
,a)))           S->a
,a)))           L'->,SL'
)))             S->a
)))             L'->^
))              L'->^
)               L'->^

 String is successfully parsed


...Program finished with exit code 0
Press ENTER to exit console.
```