LAB 6 - RECURSIVE DESCENT PARSER FOR SIMPLE GRAMMARS

Name : Pranamya G Kulal
Class : CSE A
Roll no: 8
Reg no: 220905018

Q1)S → a | > | (T)
T → T, S|S

i) Grammar without left recusrion

S → a | > | (T)
T → ST'
T'-> ,ST'| epsilon

ii) Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int curr = 0;
char str[100];

void S();
void T();
void Tprime();

void invalid() {
   printf("-----------------ERROR-----------------\n");
   exit(0);
}
void valid(){
   printf("-----------------SUCCESS-----------------\n");
   exit(0);
}
void S(){
   if(str[curr] == 'a'){
      curr++;
      return;
   }
   else if(str[curr] == '>'){
      curr++;
      return;
   }
   else if(str[curr] == '('){
      curr++;
      T();
      if(str[curr] == ')'){
         curr++;
         return;
      }
      else invalid();
   }
```

```
    else invalid();
}

void T(){
   S();
   Tprime();
}

void Tprime(){
    if(str[curr] == ','){
        curr++;
        S();
        Tprime();
    }
}

int main(){
    printf("Enter string: ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else invalid();
}
```

iii) Terminal

```
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ gcc -o l6q1 l6q1.c
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q1
Enter string: (a,a)$
-----------------SUCCESS-----------------
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q1
Enter string: (a)
-----------------ERROR-------------------
```

Q2) S → UVW
U → (S) | aSb | d
V → aV | epsilon
W → cW | epsilon

No left recusrion
i) Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int curr = 0;
char str[100];
void S();
void U();
void V();
void W();

void invalid() {
    printf("-----------------ERROR-------------------\n");
```

```
    exit(0);
}
void valid(){
    printf("----------------------------SUCCESS--------------------\n");
    exit(0);
}
void S(){
    U();
    V();
    W();
}
void U(){
    if(str[curr] == '('){
        curr++;
        S();
        if(str[curr] == ')'){
            curr++;
            return;
        }else invalid();
    }
    else if(str[curr] == 'a'){
        curr++;
        S();
        if(str[curr] == 'b'){
            curr++;
            return;
        }else invalid();
    }
    else if(str[curr] == 'd') {
        curr++;
        return;
    }
    else invalid();
}
void V(){
    if(str[curr] == 'a'){
        curr++;
        V();
    }
}
void W(){
    if(str[curr] == 'c'){
        curr++;
        W();
    }
}
int main(){
    printf("Enter string: ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else invalid();
```

}

ii) Terminal

```
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ gcc -o l6q2 l6q2.c
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q2
Enter string: dac$
----------------------------SUCCESS--------------------
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q2
Enter string: (ac)$
-----------------ERROR-------------------
```

Q3) S → aAcBe
A → Ab|b
B → d

i) Grammar without left recusrion

S → aAcBe
A → bA'
A'-> bA'| epsilon
B → d

ii) Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int curr = 0;
char str[100];
void S();
void A();
void Aprime();
void B();

void invalid() {
   printf("-----------------ERROR-------------------\n");
   exit(0);
}
void valid(){
   printf("----------------------------SUCCESS--------------------\n");
   exit(0);
}

void S(){
   if(str[curr] == 'a'){
      curr++;
      A();
      if(str[curr] == 'c'){
         curr++;
         B();
         if(str[curr] == 'e'){
            curr++;
            return;
         }else invalid();
```

```
        }else invalid();
    }else invalid();
}

void A(){
    if(str[curr] == 'b'){
        curr++;
        Aprime();
    }else invalid();
}

void Aprime(){
    if(str[curr] == 'b'){
        curr++;
        Aprime();
    }
}

void B(){
    if(str[curr] == 'd'){
        curr++;
        return;
    }
}
int main(){
    printf("Enter string: ");
    scanf("%s", str);
    S();
    if(str[curr] == '$') valid();
    else invalid();
}
```

ii) Terminal

```
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ gcc -o l6q3 l6q3.c
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q3
Enter string: abbcde$
----------------------------SUCCESS--------------------
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q3
Enter string: accd$
-----------------ERROR-------------------
```

Q4) S →(L) | a
L → L,S | S

i) Grammar without left recursion

S →(L) | a
L → SL'
L' -> ,SL' | epsilon

ii) Code

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <string.h>
int curr = 0;
char str[100];

void S();
void L();
void Lprime();

void invalid() {
   printf("-----------------ERROR-------------------\n");
   exit(0);
}
void valid(){
   printf("-----------------SUCCESS-------------------\n");
   exit(0);
}

void S(){
   if(str[curr] == '('){
      curr++;
      L();
      if(str[curr] == ')'){
         curr++;
         return;
      }else invalid();
   }
   else if(str[curr] == 'a'){
      curr++;
      return;
   }
   else invalid();
}

void L(){
   S();
   Lprime();
}

void Lprime(){
   if(str[curr] == ','){
      curr++;
      S();
      Lprime();
   }
}

int main(){
   printf("Enter string: ");
   scanf("%s", str);
   S();
   if(str[curr] == '$') valid();
   else invalid();
```

```
}
```

ii) Terminal

```
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ gcc -o l6q4 l6q4.c
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q4
Enter string: (a,a)$
----------------SUCCESS--------------------
CD_LAB_A1@debianpc-02:~/Desktop/220905018/Lab6-RDP$ ./l6q4
Enter string: aaa$
----------------ERROR-------------------
```