

NAME: AYANIK PAUL

Roll No. 22

D1

LAB 6

Write a recursive descent parser for the following simple grammars.

1. $S \rightarrow a \mid > \mid (T)$
 $T \rightarrow T, S \mid S$

After removing Left Recursion ,

$S \rightarrow a \mid > \mid (T)$

$T \rightarrow S T'$

$T' \rightarrow , S T' \mid \epsilon$

CODE:

```
q1.c x #include< q2.c x q
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 int curr = 0;
5 char str[100];
6 void S();
7 void T();
8 void Tprime();
9 void invalid();
10 void valid();
11 void invalid() {
12     printf("-----ERROR!-----\n");
13     exit(0);
14 }
15 void valid() {
16     printf("-----SUCCESS!-----\n");
17     exit(0);
18 }
19 // S -> a | > | ( T )
20 void S() {
21     if (str[curr] == 'a') {
22         curr++;
23         return;
24     }
25     else if (str[curr] == '>') {
26         curr++;
27         return;
28     }
29     else if (str[curr] == '(') {
30         curr++;
31         T();
32         if (str[curr] == ')') {
33             curr++;
34             return;
35         } else {
36             invalid();
37         }
38     } else {
39         invalid();
40     }
41 }
```

```

28     }
29     else if (str[curr] == '(') {
30         curr++;
31         T();
32         if (str[curr] == ')') {
33             curr++;
34             return;
35         } else {
36             invalid();
37         }
38     } else {
39         invalid();
40     }
41 }
42 // T -> S T'
43 void T() {
44     S();
45     Tprime();
46 }
47 // T' -> , S T' | ε
48 void Tprime() {
49     if (str[curr] == ',') {
50         curr++;
51         S();
52         Tprime();
53     }
54     // Epsilon
55 }
56
57 int main() {
58     printf("Enter String: ");
59     scanf("%s", str);
60     S();
61     if (str[curr] == '\0') {
62         valid();
63     } else {
64         invalid();
65     }
66     return 0;
67 }
68

```

OUTPUT:

```

student@oslab-02:~/220905128/lab6$ cc q1.c
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: (A,A)
-----ERROR!-----
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: (a,a)
-----SUCCESS!-----

```

Q2)

$S \rightarrow UVW$

$U \rightarrow (S) \mid aSb \mid d$

$V \rightarrow aV \mid$

$W \rightarrow cW \mid$

CODE:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int curr = 0;
5  char str[100];
6  void S();
7  void U();
8  void V();
9  void W();
10 void invalid();
11 void valid();
12 void invalid() {
13     printf("-----ERROR!-----\n");
14     exit(0);
15 }
16 void valid() {
17     printf("-----SUCCESS!-----\n");
18     exit(0);
19 }
20 // S -> UVW
21 void S() {
22     U();
23     V();
24     W();
25 }
26 // U -> (S) | aSb | d
27 void U() {
28     if (str[curr] == '(') {
29         curr++;
30         S();
31         if (str[curr] == ')') {
32             curr++;
33             return;
34         } else {
35             invalid();
36         }
37     } else if (str[curr] == 'a') {
38         curr++;
39         S();
40         if (str[curr] == 'b') {
41             curr++;
```

```

40         if (str[curr] == 'b') {
41             curr++;
42             return;
43         } else {
44             invalid()
45         }
46     } else if (str[curr] == 'd') {
47         curr++;
48         return;
49     } else {
50         invalid();
51     }
52 }
53 // V -> aV |Epsilon
54 void V() {
55     if (str[curr] == 'a') {
56         curr++;
57         V();
58     }
59     //Epsilon
60 }
61
62 // W -> cW | Epsilon
63 void W() {
64     if (str[curr] == 'c') {
65         curr++;
66         W();
67     }
68     //Epsilon
69 }
70 int main() {
71     printf("Enter String: ");
72     scanf("%s", str);
73     S();
74     if (str[curr] == '\0') {
75         valid();
76     } else {
77         invalid();
78     }
79     return 0;
80 }

```

OUTPUT:

```

student@oslab-02:~/220905128/lab6$ cc q2.c
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: (adb)
-----SUCCESS!-----
student@oslab-02:~/220905128/lab6$

```

Q3)

$S \rightarrow aAcBe$

$A \rightarrow Ab|b$

$B \rightarrow d$

After removing Left Recursion ,

$S \rightarrow aAcBe$

$A \rightarrow b A'$

$A' \rightarrow b A' | \epsilon$

$B \rightarrow d$

CODE:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int curr = 0;
5  char str[100];
6  void S();
7  void A();
8  void A_prime();
9  void B();
10 void invalid();
11 void valid();
12 void invalid() {
13     printf("-----ERROR!-----\n");
14     exit(0);
15 }
16 void valid() {
17     printf("-----SUCCESS!-----\n");
18     exit(0);
19 }
20 // S -> aAcBe
21 void S() {
22     if (str[curr] == 'a') {
23         curr++;
24         A();
25         if (str[curr] == 'c') {
26             curr++;
27             B();
28             if (str[curr] == 'e') {
29                 curr++;
30                 return;
31             } else {
32                 invalid();
33             }
34         } else {
35             invalid();
36         }
37     } else {
38         invalid();
39     }
40 }
41 // A -> b A'
```

```

41 // A -> b A'
42 void A() {
43     if (str[curr] == 'b') {
44         curr++;
45         A_prime();
46     } else {
47         invalid();
48     }
49 }
50 // A' -> b A' | ε
51 void A_prime() {
52     if (str[curr] == 'b') {
53         curr++;
54         A_prime();
55     }
56 }
57 // B -> d
58 void B() {
59     if (str[curr] == 'd') {
60         curr++;
61         return;
62     } else {
63         invalid();
64     }
65 }
66 int main() {
67     printf("Enter String: ");
68     scanf("%s", str);
69     S();
70     if (str[curr] == '\0') {
71         valid();
72     } else {
73         invalid();
74     }
75     return 0;
76 }

```

OUTPUT:

```

student@oslab-02:~/220905128/lab6$ cc q3.c
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: abcde
-----SUCCESS!-----
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: abbbcd
-----SUCCESS!-----
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: abbce
-----ERROR!-----

```


Q4)

$S \rightarrow (L) \mid a$

$L \rightarrow L,S \mid S$

After removing Left Recursion ,

$S \rightarrow (L) \mid a$

$L \rightarrow S L'$

$L' \rightarrow , S L' \mid \text{Epsilon}$

CODE:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int curr = 0;
5  char str[100];
6  void S();
7  void L();
8  void L_prime();
9  void invalid();
10 void valid();
11 void invalid() {
12     printf("-----ERROR!-----\n");
13     exit(0);
14 }
15 void valid() {
16     printf("-----SUCCESS!-----\n");
17     exit(0);
18 }
19 // S -> (L) | a
20 void S() {
21     if (str[curr] == '(') {
22         curr++;
23         L();
24         if (str[curr] == ')') {
25             curr++;
26             return;
27         } else {
28             invalid();
29         }
30     } else if (str[curr] == 'a') {
31         curr++;
32         return;
33     } else {
34         invalid();
35     }
36 }
37 // L -> S L'
38 void L() {
39     S();
40     L_prime();
41 }
```

```

41 }
42 // L' -> , S L' |Epsilon
43 void L_prime() {
44     if (str[curr] == ',') {
45         curr++;
46         S();
47         L_prime();
48     }
49 }
50 int main() {
51     printf("Enter String: ");
52     scanf("%s", str);
53     S();
54     if (str[curr] == '\0') {
55         valid();
56     } else {
57         invalid();
58     }
59     return 0;
60 }

```

OUTPUT:

```

student@oslab-02:~/220905128/lab6$ cc q4.c
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: (a,a)
-----SUCCESS!-----
student@oslab-02:~/220905128/lab6$ ./a.out
Enter String: (a,a,a)
-----SUCCESS!-----

```