LAB NO: 6 RECURSIVE DESCENT PARSER FOR SIMPLE GRAMMARS

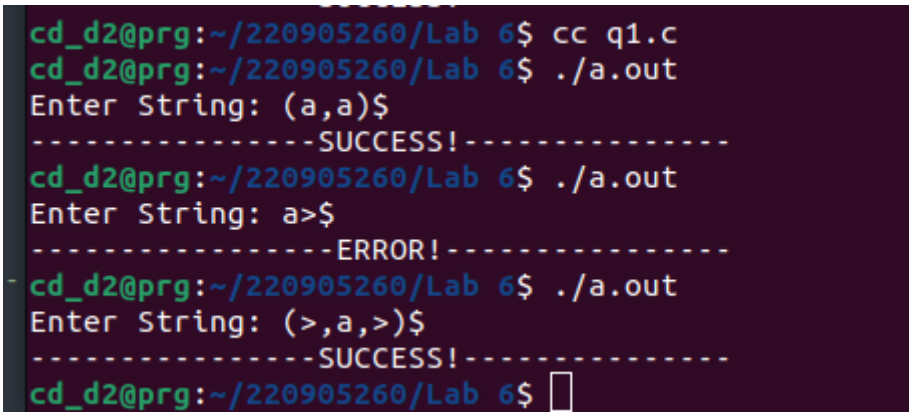
Name: Ketan Goud
Reg No: 220905260
Roll No: 39
Section: D D2

Write a recursive descent parser for the following simple grammars.
1. S→a | > | (T)
T→T, S|S

```
/*
S→ a | > | ( T )
T→ T,S | S
*/

/*
S-> a | > | ( T )
T-> ST'
T'-> ,ST' |
*/
#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 T()
{
        S();
        Tprime();
}
void Tprime()
{
        if(str[curr]==',')
        {
                curr++;
                S();
                Tprime();
```

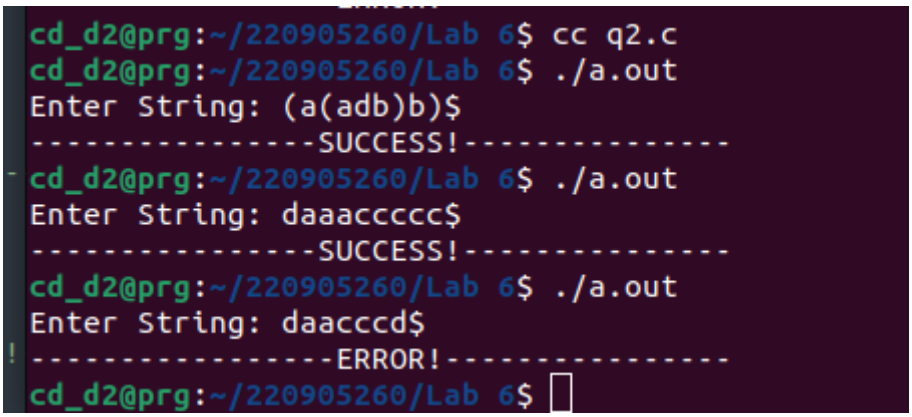
```
        }
}
void S()
{
        if(str[curr]=='a')
        {
                curr++;
                return;
        }
        else if(str[curr]=='>')
        {
                curr++;
                return;
        }
        else if(str[curr] == '(')
        {
                curr++;
                T();
                if(str[curr] == ')')
                {
                        curr++;
                        return;
                }
                else
                invalid();
        }
        else
                invalid();
}
int main()
{
        printf("Enter String: ");
        scanf("%s", str);
        S();
        if(str[curr] == '$')
        valid();
        else
        // printf("%c\n", str[curr]);
        invalid();
}
```



2. S →UVW
U →(S) | aSb | d
V → aV |
W → cW |

```
/*
S→UVW
U →(S) | aSb | d
V → aV |
W → cW |
*/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int curr = 0;
char str[100];
void S();
void V();
void U();
void W();
void invalid()
{
	printf("-----------------ERROR!----------------\n");
	exit(0);
}
void valid()
{
	printf("----------------SUCCESS!---------------\n");
	exit(0);
}
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();
        }
}
void S()
{
        U();
        V();
        W();
}
int main()
{
        printf("Enter String: ");
        scanf("%s", str);
        S();
        if(str[curr] == '$')
        valid();
        else
        // printf("%c\n", str[curr]);
        invalid();
}
```



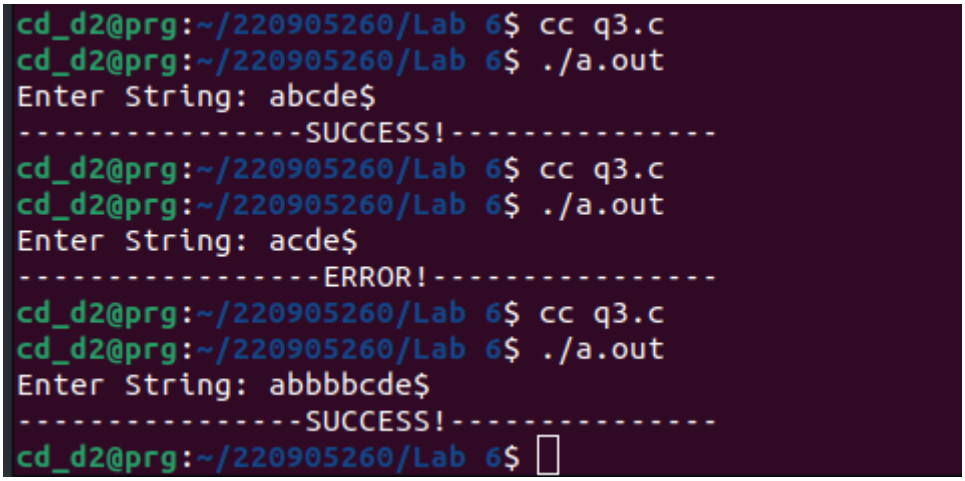
3. S → aAcBe
A → Ab|b
B → d

```
/*
S → aAcBe
A → Ab|b
B → d
*/
/*
S -> aAcBe
A -> bA'
A'-> bA' |
B -> d
*/

#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
        // printf("%c\n", str[curr]);
        invalid();
}
```



4. S →(L) | a
L → L,S | S

```
/*
S →(L) | a
L → L,S | S
*/
/*
S -> (L) | a
L -> SL'
L'-> ,SL' |
*/

#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 L()
{
	S();
	Lprime();
}
void Lprime()
{
	if(str[curr]==',')
	{
		curr++;
		S();
		Lprime();
	}
}
void S()
{
	if(str[curr]=='(')
	{
		curr++;
		L();
		if(str[curr]==')')
		{
			curr++;
			return;
		}
		else
		{
			invalid();
		}
	}
	else if(str[curr]=='a')
	{
		curr++;
		return;
	}
	else
	{
		invalid();
	}
}
int main()
{
	printf("Enter String: ");
	scanf("%s", str);
	S();
```

```
        if(str[curr] == '$')
        valid();
        else
        // printf("%c\n", str[curr]);
        invalid();
}
```

```
cd_d2@prg:~/220905260/Lab 6$ cc q4.c
cd_d2@prg:~/220905260/Lab 6$ ./a.out
Enter String: (a,a)$
----------------SUCCESS!---------------
cd_d2@prg:~/220905260/Lab 6$ ./a.out
Enter String: a(a,a)$
-----------------ERROR!----------------
cd_d2@prg:~/220905260/Lab 6$ ./a.out
Enter String: ((a),(a))$
----------------SUCCESS!---------------
cd_d2@prg:~/220905260/Lab 6$
```