

Exercise 3

Q : Construct a recursive descent parser in C for the grammar G_2 ?

Solution

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

char input[100];
int i = 0;           // current position in input
char lookahead;      // current character

void advance() {
    i++;
    if (input[i] != '\0')
        lookahead = input[i];
    else
        lookahead = '$'; // End of input marker
}

void match(char x) {
    if (lookahead == x)
        advance();
    else {
        printf("Expected '%c'", x);
    }
}

/* -----
Grammar:
S → a B C d | d C B e
B → b B | ε
C → c a | a c | ε
-----*/

void S();
void B();
void C();

/* S → a B C d | d C B e */
void S() {
    if (lookahead == 'a') {
        match('a');
```

```

        B();
        C();
        match('d');
    }
    else if (lookahead == 'd') {
        match('d');
        C();
        B();
        match('e');
    }
    else {
        printf("Expected 'a' or 'd' to start S");
    }
}

/* B → b B | ε */
void B() {
    if (lookahead == 'b') {
        match('b');
        B();
    }
    else {
        // epsilon production
        return;
    }
}

/* C → c a | a c | ε */
void C() {
    if (lookahead == 'c') {
        match('c');
        match('a');
    }
    else if (lookahead == 'a') {
        match('a');
        match('c');
    }
    else {
        // epsilon production
        return;
    }
}

```

```
int main() {  
    printf("Enter input string: ");  
    scanf("%s", input);  
  
    lookahead = input[0];  
  
    S(); // Start parsing from the start symbol  
  
    if (lookahead == '$')  
        printf("Parsing successful.\n");  
    else  
        printf("Unexpected trailing characters");  
  
    return 0;  
}
```