13. Write a C program to implement either Top Down parsing technique or Bottom Up Parsing technique to check whether the given input string is satisfying the grammar or not.

CODE:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

char \*input;

char lookahead;

void S();

void A();

void match(char t);

void S() {

if (lookahead == 'a') {

match('a');

A();

if (lookahead == 'b') {

match('b');

} else {

printf("Syntax error: expected 'b'\n");

exit(1);

}

} else {

printf("Syntax error: expected 'a'\n");

exit(1);

}

}

void A() {

if (lookahead == 'c') {

match('c');

} // epsilon production does nothing

}

void match(char t) {

if (lookahead == t) {

lookahead = \*input++;

} else {

printf("Syntax error: expected '%c'\n", t);

exit(1);

}

}

void parse(char \*expr) {

input = expr;

lookahead = \*input++;

S();

if (lookahead == '\0') {

printf("Parsing successful\n");

} else {

printf("Syntax error: unexpected '%c'\n", lookahead);

}

}

int main() {

char expr[] = "acb"; // You can change this to test different inputs

parse(expr);

return 0;

}

