14. Implement the concept of Shift reduce parsing in C Programming?  
#include <stdio.h>

#include <string.h>

#define MAX 100

char stack[MAX][MAX]; // Stack for parsing

int top = -1;

void push(char \*str) {

if (top < MAX - 1) {

strcpy(stack[++top], str);

}

}

void pop() {

if (top >= 0) {

top--;

}

}

void displayStack() {

printf("Stack: ");

for (int i = 0; i <= top; i++) {

printf("%s ", stack[i]);

}

printf("\n");

}

void reduce() {

while (top >= 2) {

if ((strcmp(stack[top - 1], "+") == 0 || strcmp(stack[top - 1], "\*") == 0) &&

strcmp(stack[top - 2], "E") == 0 && strcmp(stack[top], "E") == 0) {

pop(); pop();

push("E");

printf("Reduced by E -> E %s E\n", stack[top - 1]);

} else if (strcmp(stack[top], "id") == 0) {

pop();

push("E");

printf("Reduced by E -> id\n");

} else if (strcmp(stack[top - 1], "(") == 0 && strcmp(stack[top + 1], ")") == 0 && strcmp(stack[top], "E") == 0) {

pop(); pop(); pop();

push("E");

printf("Reduced by E -> (E)\n");

} else {

break;

}

displayStack();

}

}

void shiftReduceParse(char \*tokens[], int n) {

printf("Starting Shift-Reduce Parsing...\n");

for (int i = 0; i < n; i++) {

push(tokens[i]);

printf("Shift: %s\n", tokens[i]);

displayStack();

reduce();

}

if (top == 0 && strcmp(stack[top], "E") == 0) {

printf("Parsing Successful: String is accepted.\n");

} else {

printf("Parsing Failed: String is not accepted.\n");

}

}

int main() {

char \*input[] = {"id", "+", "id", "\*", "id"};

int size = sizeof(input) / sizeof(input[0]);

shiftReduceParse(input, size);

return 0;

}

