Program no: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 <string.h>

#define MAX 100

char input[MAX];

char stack[MAX];

int top = -1;

void push(char c) {

stack[++top] = c;

}

char pop() {

return stack[top--];

}

int isOperator(char c) {

return (c == '+' || c == '-' || c == '\*' || c == '/');

}

int isTerminal(char c) {

return (c >= 'a' && c <= 'z');

}

void parse() {

int i = 0;

push('$');

push('E');

while (top != -1) {

char stackTop = pop();

char currentInput = input[i];

if (stackTop == currentInput) {

i++;

} else if (isOperator(stackTop)) {

if (isOperator(currentInput)) {

continue;

} else {

printf("Error: Invalid input\n");

return;

}

} else if (stackTop == 'E') {

if (currentInput == 'a') {

push('E');

push('T');

} else {

printf("Error: Invalid input\n");

return;

}

} else {

printf("Error: Invalid input\n");

return;

}

}

if (input[i] == '\0' && top == -1) {

printf("Input string is accepted\n");

} else {

printf("Input string is rejected\n");

}

}

int main() {

printf("Enter the input string: ");

scanf("%s", input);

parse();

return 0;

}

Output:

Input Output

a Input string is accepted

aa Input string is accepted

aaa Input string is accepted

.

Program no: 14

Implement the concept of Shift reduce parsing in C Programming.

Code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX 100

char stack[MAX][MAX];

int top = -1;

void push(char \*str) {

if (top < MAX - 1) {

top++;

strcpy(stack[top], str);

}

}

char\* pop() {

if (top >= 0) {

return stack[top--];

}

return NULL;

}

int isOperator(char c) {

return (c == '+' || c == '-' || c == '\*' || c == '/');

}

void shift(char \*input, int \*index) {

char temp[2] = {input[\*index], '\0'};

push(temp);

(\*index)++;

}

void reduce() {

char \*op1 = pop();

char \*operator = pop();

char \*op2 = pop();

if (op1 && operator && op2) {

char result[MAX];

sprintf(result, "(%s %s %s)", op2, operator, op1);

push(result);

}

}

void parse(char \*input) {

int index = 0;

while (input[index] != '\0') {

if (isOperator(input[index])) {

shift(input, &index);

reduce();

} else {

char temp[2] = {input[index], '\0'};

push(temp);

index++;

}

}

}

int main() {

char input[MAX];

printf("Enter the expression: ");

scanf("%s", input);

parse(input);

printf("Parsed Expression: %s\n", pop());

return 0;

}

Output:

Enter the expression: a+b\*c

Parsed Expression: c