Infix to postfix

Sunday, 10 March 2024 2:39 pm

Infix to postfix

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

#define MAX_SIZE 100

typedef struct {
    char stack[MAX_SIZE];
    int top;
} Stack;

void initialize(Stack *s) {
    s->top = -1;
}

int isEmpty(Stack *s) {
    return (s->top == -1);
}

int isFull(Stack *s) {
    return (s->top == MAX_SIZE - 1);
}

void push(Stack *s, char c) {
    if (isFull(s)) {
        printf("Stack Overflow\n");
        exit(EXIT_FAILURE);
    }
    s->stack[++(s->top)] = c;
}

char pop(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack Underflow\n");
        exit(EXIT_FAILURE);
    }
    return s->stack[(s->top)--];
}

int precedence(char op) {
    switch (op) {
        case '+':
        case '-':
            return 1;
        case '*':
        case '/':
            return 2;
        default:
            return 0;
    }
}

void infixToPostfix(char *infix, char *postfix) {
    Stack stack;
    initialize(&stack);
```

```
    int i, j = 0;
    for (i = 0; infix[i] != '\0'; i++) {
        if (isalnum(infix[i])) {
            postfix[j++] = infix[i];
        } else if (infix[i] == '(') {
            push(&stack, infix[i]);
        } else if (infix[i] == ')') {
            while (!isEmpty(&stack) && stack.stack[stack.top] != '(') {
                postfix[j++] = pop(&stack);
            }
            pop(&stack); // Discard the '('
        } else {
            while (!isEmpty(&stack) && precedence(infix[i]) <= precedence(stack.stack[stack.top])) {
                postfix[j++] = pop(&stack);
            }
            push(&stack, infix[i]);
        }
    }

    while (!isEmpty(&stack)) {
        postfix[j++] = pop(&stack);
    }
    postfix[j] = '\0';
}

int main() {
    char infix[MAX_SIZE];
    char postfix[MAX_SIZE];
                printf("name=kongara sai\nreg no=192365025\n");

    printf("Enter an infix expression: ");
    scanf("%s", infix);

    infixToPostfix(infix, postfix);

    printf("Postfix expression: %s\n", postfix);

    return 0;
}
```

```
/tmp/zUEGiRQTJ9.o
name=kongara sai
reg no=192365025
Enter an infix expression: (a+b)*c+d
Postfix expression: ab+c*d+
```