

Institute of Computer Technology  
B. Tech. Computer Science and Engineering  
Sub: DS Branch: BDA Class: A

Name: Jatin Patel  
Enrolment No.: 23162121029  
Sem: 3  
Class: A  
Subject: DS  
Practical: 06  
Date: 29/08/24

### Practical 6:

Stack infix to prefix.

Write a C/C++ program to convert infix notation to prefix notation using stack.

Input:

(a+b)\*(c+d)

Output:

Postfix notation: \*+ab+cd

#### Input Code:

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

#define MAX 100

char stack[MAX];
int top = -1;

void push(char x) {
    if (top >= MAX - 1) {
        printf("Stack overflow\n");
        return;
    }
}
```

```

    stack[++top] = x;
}

char pop() {
    if (top == -1) {
        printf("Stack underflow\n");
        return -1;
    }
    return stack[top--];
}

int precedence(char x) {
    if (x == '+' || x == '-') return 1;
    if (x == '*' || x == '/') return 2;
    return 0;
}

void reverse(char *exp) {
    int i, len = strlen(exp);
    for (i = 0; i < len / 2; i++) {
        char temp = exp[i];
        exp[i] = exp[len - i - 1];
        exp[len - i - 1] = temp;
    }
}

void infixToPrefix(char *infix, char *prefix) {
    int i, j = 0;
    reverse(infix);
    for (i = 0; infix[i]; i++) {
        if (infix[i] == '(') infix[i] = ')';
        else if (infix[i] == ')') infix[i] = '(';
    }

    for (i = 0; infix[i]; i++) {
        if (isalnum(infix[i])) {
            prefix[j++] = infix[i];
        } else if (infix[i] == '(') {
            push(infix[i]);
        } else if (infix[i] == ')') {
            while (top != -1 && stack[top] != '(') {
                prefix[j++] = pop();
            }
            pop();
        } else {
            while (top != -1 && precedence(stack[top]) >=
precedence(infix[i])) {
                prefix[j++] = pop();
            }

```

```

    }
    push(infix[i]);
}
}
while (top != -1) {
    prefix[j++] = pop();
}
prefix[j] = '\0';
reverse(prefix);
}

int main() {
    char infix[MAX], prefix[MAX];
    printf("Enter infix exp.: ");
    scanf("%s", infix);
    infixToPrefix(infix, prefix);
    printf("Prefix notation: %s\n", prefix);
    return 0;
}

```

### Output:

```

PS C:\Users\jatin\OneDrive\Desktop\Academics\SEM - 3\Practicals\DS\Practical-6> & 'c:\Users\jatin\.vscode\extensions\ms-vscode.cpp
tools-1.22.0-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-azoao03s.51c' '--stdout=Microsoft
-MIEngine-Out-50ke5jp0.fmg' '--stderr=Microsoft-MIEngine-Error-hgixprgg.z2j' '--pid=Microsoft-MIEngine-Pid-qa5myrxn.j25' '--dbgExe=
C:\MinGW\bin\gdb.exe' '--interpreter=mi'
PS C:\Users\jatin\OneDrive\Desktop\Academics\SEM - 3\Practicals\DS\Practical-6> ./infix2prefix.exe
Enter infix expression: (a+b)*(c+d)
Prefix notation: **+b+c*d
PS C:\Users\jatin\OneDrive\Desktop\Academics\SEM - 3\Practicals\DS\Practical-6>

```