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

Code:

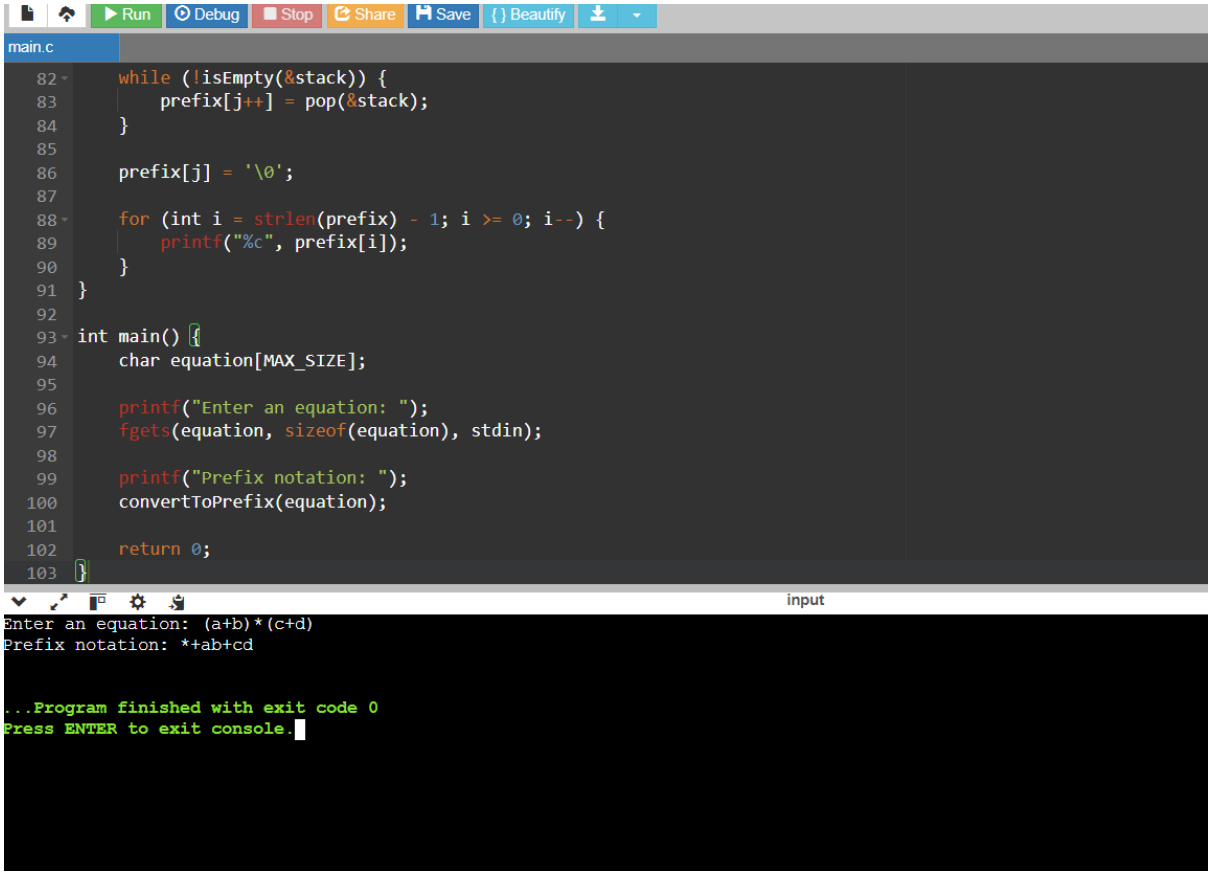
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_SIZE 100

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

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

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

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

```
void push(Stack* stack, char item) {
    if (isFull(stack)) {
        printf("Stack Overflow\n");
        exit(1);
    }
    stack->data[++stack->top] = item;
}

char pop(Stack* stack) {
    if (isEmpty(stack)) {
        printf("Stack Underflow\n");
        exit(1);
    }
    return stack->data[stack->top--];
}

int isOperator(char ch) {
    return (ch == '+' || ch == '-' || ch == '*' || ch == '/');
}

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

void convertToPrefix(char* equation) {
    int len = strlen(equation);
```

```
Stack stack;
initialize(&stack);
char prefix[MAX_SIZE];
int j = 0;

for (int i = len - 1; i >= 0; i--) {
    if (equation[i] == ' ' || equation[i] == '\t')
        continue;

    if (isOperator(equation[i])) {
        while (!isEmpty(&stack) && precedence(stack.data[stack.top]) > precedence(equation[i])) {
            prefix[j++] = pop(&stack);
        }
        push(&stack, equation[i]);
    } else if (equation[i] == ')') {
        push(&stack, equation[i]);
    } else if (equation[i] == '(') {
        while (!isEmpty(&stack) && stack.data[stack.top] != ')') {
            prefix[j++] = pop(&stack);
        }
        if (!isEmpty(&stack) && stack.data[stack.top] == ')') {
            pop(&stack);
        }
    } else {
        prefix[j++] = equation[i];
    }
}

while (!isEmpty(&stack)) {
    prefix[j++] = pop(&stack);
}
```

```
    prefix[j] = '\0';

    for (int i = strlen(prefix) - 1; i >= 0; i--) {
        printf("%c", prefix[i]);
    }
}

int main() {
    char equation[MAX_SIZE];

    printf("Enter an equation: ");
    fgets(equation, sizeof(equation), stdin);

    printf("Prefix notation: ");
    convertToPrefix(equation);

    return 0;
}
```

Output:


Run
Debug
Stop
Share
Save
{ } Beautify
main.c
while (!isEmpty(&stack)) {
prefix[j++] = pop(&stack);
}
prefix[j] = '\0';
for (int i = strlen(prefix) - 1; i >= 0; i--) {
printf("%c", prefix[i]);
}
}
int main() {
char equation[MAX_SIZE];
printf("Enter an equation: ");
fgets(equation, sizeof(equation), stdin);
printf("Prefix notation: ");
convertToPrefix(equation);
return 0;
}
input
Enter an equation: (a+b)*(c+d)
Prefix notation: *+ab+cd
...Program finished with exit code 0
Press ENTER to exit console.