22_HEMANT_18.c

```c
/*
Roll no : 22
Batch: A
Author name: Hemant Gupta
Date: 23/08/2024
Description: Program to convert infix to postfix expression

*/


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

#define MAX 100

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

// Push an item onto the stack
void push(char c) {
    stack[++top] = c;
}

// Pop an item from the stack
char pop() {
    return stack[top--];
}

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

// Convert infix to postfix
void infixToPostfix(char* infix, char* postfix) {
    int i = 0, j = 0;

    while (infix[i] != '\0') {
        if (isalnum(infix[i])) { // If operand, add to postfix
            postfix[j++] = infix[i];
        } else if (infix[i] == '(') {
            push(infix[i]);
        } else if (infix[i] == ')') {
            while (top != -1 && stack[top] != '(') { // Check if stack is not empty
                postfix[j++] = pop();
            }
            pop(); // Remove '('
        } else { // Operator
```

```c
            while (top != -1 && precedence(stack[top]) >= precedence(infix[i])) { // Check if stack is not empty
                postfix[j++] = pop();
            }
            push(infix[i]);
        }
        i++;
    }

    // Pop remaining operators
    while (top != -1) { // Check if stack is not empty
        postfix[j++] = pop();
    }
    postfix[j] = '\0'; // Null terminate
}

int main() {
    char infix[MAX], postfix[MAX];

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

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

    return 0;
}

```