Implement Infix to Postfix Expression Conversion using Stack

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

char stack[20];
int top = -1;

void push(char x) {
	stack[++top] = x;
}

char pop() {
	if (top == -1)
		return -1;
	else
		return stack[top--];
}

int checkpriority(char x) {
	if (x == '(')
		return 0;
	if (x == '+' || x == '-')
		return 1;
	if (x == '/' || x == '*')
		return 2;
	if (x == '^')
		return 3;
	return -1;
}

int main() {
	char expression[100];
	char *e, x;
	printf("Please enter any expression:");
	scanf("%s", expression);
	e = expression;
	printf("Postfix expression: ");
	while (*e != '\0') {
		if (isalnum(*e)) {
			printf("%c", *e);
		} else if (*e == '(') {
			push(*e);
		} else if (*e == ')') {
			while ((x = pop()) != '('
```

```
                printf("%c", x);
        } else {
            while (top != -1 && checkpriority(stack[top]) >= checkpriority(*e))
                printf("%c", pop());
            push(*e);
        }
        e++;
    }
    while (top != -1) {
        printf("%c", pop());
    }
    printf("\n");
    return 0;
}
```

A+b*c-d/e*f ------- **A b c * + d e / f * -**
(A+b*c-d)/(E*f) ------ **A b c * + d - E f * /**
A*b+c*d-e*f --------- **A b * c d * + e f * -**
A+b+C+d+e+f+g ------- **A b + C + d + e + f + g +**