

4. Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, (Remainder), ^ (Power) and alphanumeric operands.

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <ctype.h>

#define MAX 100

// Function to return precedence of operators

int precedence(char ch) {

switch (ch) {

    case '+':

    case '-':

        return 1;

    case '*':

    case '/':

    case '%':

        return 2;

    case '^':

        return 3;

    }

    return -1;

}

// Function to convert infix expression to postfix expression
```

```

void infixToPostfix(char* infix, char* postfix) {
    int i, j;

    char stack[MAX];

    int top = -1;

    for (i = 0, j = 0; infix[i]; ++i) {
        // If the scanned character is an operand, add it to output.
        if (isalnum(infix[i]))
            postfix[j++] = infix[i];

        // If the scanned character is an '(', push it to the stack.
        else if (infix[i] == '(')
            stack[++top] = infix[i];

        // If the scanned character is an ')', pop and output from the stack
        // until an '(' is encountered.
        else if (infix[i] == ')') {
            while (top != -1 && stack[top] != '(')
                postfix[j++] = stack[top--];

            if (top != -1)
                top--; // pop '('
        } else { // operator encountered

            while (top != -1 && precedence(infix[i]) <= precedence(stack[top]))
                postfix[j++] = stack[top--];

            stack[++top] = infix[i];
        }
    }
}

```

```
}  
  
// Pop all the remaining elements from the stack  
while (top != -1) {  
    postfix[j++] = stack[top--];  
}  
postfix[j] = '\0';  
}  
  
int main() {  
    char infix[MAX];  
    char postfix[MAX];  
    printf("Enter infix expression: ");  
    gets(infix);  
  
    infixToPostfix(infix, postfix);  
    printf("Postfix expression: %s\n", postfix);  
    return 0;  
}
```