(Q) Demonstrate Infix to postfix conversion.

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#define MAX_SIZE 100

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

void infixToPostfix(char* exp){
    char postfix[MAX_SIZE];
    int i,j;
    int stack[MAX_SIZE];
    int top = -1;

    for(i=0, j=0 ;exp[i] != '\0' ; i++){
    if(isalnum(exp[i]))
        postfix[j++] = exp[i];
    else if(exp[i] == '(')
        stack[++top] = exp[i];
    else if(exp[i] == ')'){
    while (top!= -1 && stack[top] != '(')
        postfix[j++] = stack[top--];
    if (top == -1){
    printf("Invalid Expression\n");
    return;
    }
    top--;
    }
    else{
    while(top != -1 && precedence(stack[top]) >= precedence(exp[i]))
        postfix[j++] = stack[top--];
    stack[++top] = exp[i];
    }
    }

    while(top != -1){
    if(stack[top] == '('){
    printf("Invalid Expression");
    return;
       }
       postfix[j++]  = stack[top--];
    }
```

```
    postfix[j] = '\0';
    printf("Postfix: %s\n",postfix);
}

int main(){
    char exp[MAX_SIZE];
    printf("Enter the infix expression: ");
    fgets(exp, sizeof(exp), stdin);

    if(exp[strlen(exp) - 1] == '\n')
        exp[strlen(exp) - 1] == '\0';
    infixToPostfix(exp);
    return 0;
}
```

OUTPUT :-

```
Enter the infix expression: a+b*c-(d/e+f)
Postfix: abc*+de/f+-


Process returned 0 (0x0)   execution time : 18.320 s
Press any key to continue.
```