EUROPEAN UNIVERSITY OF LEFKE
Faculty of Engineering
Department of Software Engineering

COMP 217

DATA STRUCTURES

Lab Work No. 5

Prepared by Abdelrahman Mohamed Radwan Mostafa

Student Number : 21140036

Submitted to Dr. Ferhun Yorgancıoğlu

Code :

```c
#include <stdio.h>
#define SIZE 16

typedef struct {
    char items[SIZE];
    int top;
} stack;

void init_stack(stack *s) {
    s->top = -1;
}

int is_empty(stack *s) {
    return s->top == -1;
}

int is_full(stack *s) {
    return s->top == SIZE - 1;
}

char top_of_stack(stack *s) {
    return is_empty(s) ? '\0' : s->items[s->top];
}

void push(stack *s, char c) {
    if (is_full(s)) {
        printf("Overflow: stack is full!\n");
        return;
    }
    s->top++;
    s->items[s->top] = c;
}

char pop(stack *s) {
    if (is_empty(s)) {
        printf("Underflow: stack is empty!\n");
        return '\0';
    }
```

```
    char temp = s->items[s->top];
    s->top--;
    return temp;
}

int priority(char c) {
    if (c == '$') return 3;
    else if (c == '*' || c == '/') return 2;
    else if (c == '+' || c == '-') return 1;
    else return 0;
}

int main() {
    char buf[32], output[32], *marker = buf;
    int i = 0;
    stack operator;
    init_stack(&operator);

    printf("Enter an infix expression : ");
    gets(buf);

    while (*marker != '\0') {
        while (*marker == ' ') marker++;

        if (*marker >= '0' && *marker <= '9') output[i++] = *marker;
        else if (*marker == '(') push(&operator, *marker);
        else if (*marker == '+' || *marker == '-' || *marker == '*'
|| *marker == '/' || *marker == '$') {
            while (!is_empty(&operator) && top_of_stack(&operator)
!= '(' && priority(*marker) <= priority(top_of_stack(&operator))) {
                output[i++] = pop(&operator);
            }
            push(&operator, *marker);
        }
        else if (*marker == ')') {
            while (top_of_stack(&operator) != '(') output[i++] =
pop(&operator);
            pop(&operator);
        }
```

```
        marker++;
    }

    while (!is_empty(&operator)) output[i++] = pop(&operator);

    output[i] = '\0';

    printf("The postfix form is : \n%s", output);
    return 0;
}
```

Result :