5/10/2020

classmate Date Page

LAB 3 – Infix to Postfix

WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression should consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).

```
#include <stdio.h>
#include <string.h>
int A(char symbol)
{
    switch (symbol)
    {
        case '+':
        case '-': return 2;
        case '*':
        case '/': return 4;
        case '^':
        case '$': return 5;
        case '(': return 0;
        case '#': return -1;
        default : return 8;
    }
}
int B (char symbol)
{
    switch (symbol)
    {
        case '+':
        case '-': return 1;
```

```
        case '*':
        case '/' : return 3;
        case '^' :
        case '$' : return 6;
        case '(' : return 9;
        case ')' : return 0;
        default : return 7;
    }
}
void infix_postfix(char infix[], char postfix[])
{
    int top, j, i;
    char s[30], symbol;
    top = -1;
    s[++top] = '#';
    j = 0
    for (i = 0; i < strlen(infix); i++)
    {
     symbol = infix[i];
     while (A(s[top]) > B(symbol))
     {
      postfix[j] = s[top--];
     } j++;
    if(A(s[top]) != B(symbol))
    st++top] = symbol;
    else
    top--;
}
```

```
    while (stack[top] != '#')
    {
      postfix[j++] = stack[top--];
    }
    postfix[j] = '\0';
}
void main()
{
    char infix[25];
    char infix[25];
    printf("Enter the infix expression: ");
    scanf("%s", infix);
    infix_postfix(infix, postfix);
    printf("The postfix expression is: ");
    printf("%s\n", postfix);
}
```