Shreshtha Aggarwal
IBM19CS155

Date
Page 6

9-10-20 Tab 3

Q-> WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators +, -, * and /.

Sol 2

```
#include <stdio.h>
#include <string.h>
int f (char symbol)
{
    switch (symbol)
    {
      case '+' :
      case '-' : return 2;
      case '*' :
      case '/' : return 4;
      case '^' : return 5;
      case '$' : return 0;
      case '#' : return -1;
      default : return 8;
    }
}
int g (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 infin_postfix (char infin[], char postf
                                    [])
{
    int top, i, j;
    char s[30], symbol;
    top = -1;
    s[++top] = '#';
    j = 0;
    for (i = 0; i < strlen (infin); i++)
    {
        symbol = infin[i];
        while (f(s[top]) > g(symbol))
        {
            postfin[j] = s[top--];
```

```
        j++;
    }
    if (f(s[top]) != g(symbol))
        s[++top] = symbol;
    else
        top--;
    }
    while (s[top] != '#')
    {
        postfix[j++] = s[top--];
    }
    postfix[j] = '\0';
}
int main()
{
    char infix[20];
    char postfix[20];
    printf("Enter a valid infix expression:\n");
    scanf("%s", infix);
    infix_postfix(infix, postfix);
    printf("The postfix expression:\n");
    printf("%s\n", postfix);
    return 0;
}
```

Expected Output:

Enter a valid infix expression:

A + B ^ C * D / F - G

The postfix expression:

A B C ^ D * F / + G -