Date / /
Page
SPLASH

Lab Program - 2

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

Soln.

```
#include <stdio.h>
#define N 100
int stack[N];
int top = -1;

void push(int item) {
    if (top == N-1)
        printf("Stack overflow!\n");
    else
        stack[++top] = item;
}
int pop() {
    if (top == -1)
        printf("Stack underflow!\n");
    else
        return stack[top--];
}
int priority(char op) {
    switch(op) {
    case '*': return 2;
              break;
    case '/': return 2;
              break;
```

```
        case '+' : return 1;
                   break;
        case '-' : return 1;
                   break;
        case '(' : return 0;
                   break;
    }
}
int main()
{
    char s[50];
    char t[50];
    int i;
    int choice = 1;
    do {
        i = 0;
        printf("Enter your infix expression:
               ");
        scanf("%s", s);
        for (int i=0; s[i] != '\0'; i++) {
          switch (s[i]) {
            case '(' : push('(');
                       break;
            case ')' : while (stack[top] != '(')
              {
                t[i++] = pop();
              }
              pop();
              break;
            case '*';
            case '/' :
```

```
        case '+':
        case '-': while (top != -1 &&
                  priority (stack[top]) >=
                  priority (s[i])) {
                    t[j++] = pop ();
                  }
                  push (s[i]);
                  break;
        default : t[j++] = s[i];
        }
    }
    while (top != -1) {
        t[j++] = pop ();
    }
    t[j] = '\0';
    printf ("Postfix expression for %s
            is %s . \n", s, t);
    printf ("\n :: Menu :: \n1 try
            another infix expression. \n2.
            Exit \n Enter your choice : ");
    scanf ("%d", &choice);
} while (choice != 2);
return 0;
}
```