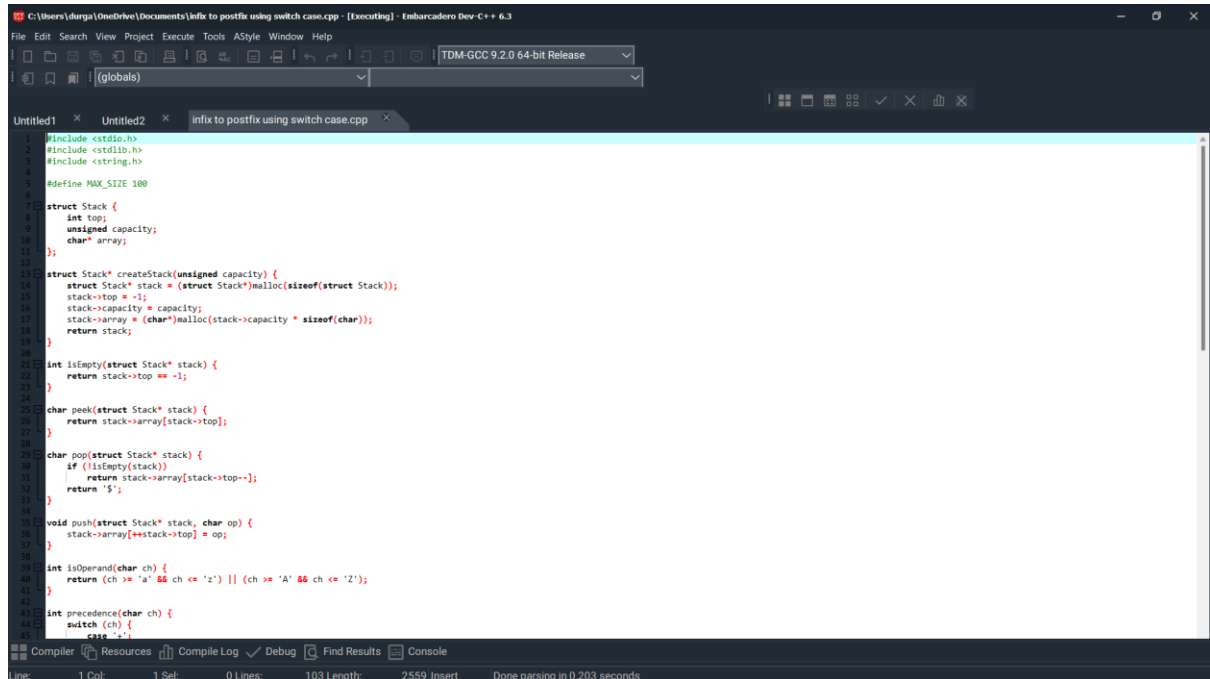
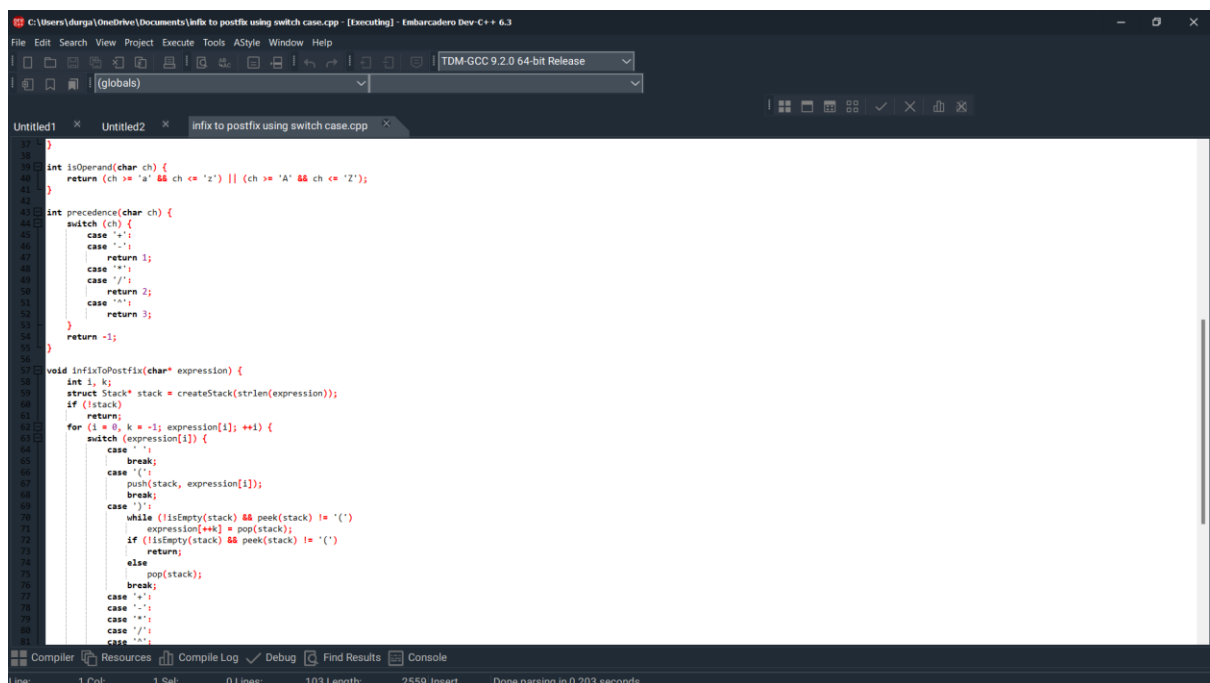


Day 5

1) infix to postfix



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define MAX_SIZE 100
6
7 struct Stack {
8     int top;
9     unsigned capacity;
10    char* array;
11};
12
13 struct Stack* createStack(unsigned capacity) {
14    struct Stack* stack = (struct Stack*)malloc(sizeof(struct Stack));
15    stack->top = -1;
16    stack->capacity = capacity;
17    stack->array = (char*)malloc(stack->capacity * sizeof(char));
18    return stack;
19}
20
21 int isEmpty(struct Stack* stack) {
22    return stack->top == -1;
23}
24
25 char peek(struct Stack* stack) {
26    return stack->array[stack->top];
27}
28
29 char pop(struct Stack* stack) {
30    if (!isEmpty(stack))
31        return stack->array[stack->top--];
32    return '$';
33}
34
35 void push(struct Stack* stack, char op) {
36    stack->array[++stack->top] = op;
37}
38
39 int isOperand(char ch) {
40    return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z');
41}
42
43 int precedence(char ch) {
44    switch (ch) {
45        case '+':
46            return 1;
47        case '*':
48            return 2;
49        case '/':
50            return 3;
51        case '^':
52            return 4;
53    }
54    return -1;
55}
```



```
56
57
58 int isOperand(char ch) {
59    return (ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z');
60}
61
62 int precedence(char ch) {
63    switch (ch) {
64        case '+':
65            return 1;
66        case '*':
67            return 2;
68        case '/':
69            return 3;
70        case '^':
71            return 4;
72    }
73    return -1;
74}
75
76 void infixToPostfix(char* expression) {
77    int i, k;
78    struct Stack* stack = createStack(strlen(expression));
79    if (!stack)
80        return;
81    for (i = 0, k = -1; expression[i]; ++i) {
82        switch (expression[i]) {
83            case ' ':
84                break;
85            case '(':
86                push(stack, expression[i]);
87                break;
88            case ')':
89                while (!isEmpty(stack) && peek(stack) != '(')
90                    expression[++k] = pop(stack);
91                if (!isEmpty(stack) && peek(stack) != '(')
92                    return;
93                else
94                    pop(stack);
95                break;
96            case '+':
97            case '*':
98            case '/':
99            case '^':
100                while (!isEmpty(stack) && precedence(peek(stack)) >= precedence(ch))
101                    expression[++k] = pop(stack);
102                push(stack, ch);
103                break;
104        }
105    }
106    while (!isEmpty(stack))
107        expression[++k] = pop(stack);
108    expression[k] = '\0';
109    printf("%s\n", expression);
110}
```

```
C:\Users\durga\OneDrive\Documents\infix to postfix using switch case.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
infix to postfix using switch case.cpp
19 if (!stack)
20 | return;
21 for (i = 0, k = -1; expression[i]; ++i) {
22     switch (expression[i]) {
23     case ' ':
24         break;
25     case '(':
26         push(stack, expression[i]);
27         break;
28     case ')':
29         while (!isEmpty(stack) && peek(stack) != '(')
30             expression[++k] = pop(stack);
31         if (!isEmpty(stack) && peek(stack) != '(')
32             return;
33         else
34             pop(stack);
35         break;
36     case '+':
37     case '-':
38     case '*':
39     case '/':
40         while (!isEmpty(stack) && precedence(expression[i]) <= precedence(peek(stack)))
41             expression[++k] = pop(stack);
42         push(stack, expression[i]);
43         break;
44     default:
45         expression[++k] = expression[i];
46     }
47 }
48 while (!isEmpty(stack))
49     expression[++k] = pop(stack);
50 expression[++k] = '\0';
51
52 int main() {
53     char infix[MAX_SIZE], postfix[MAX_SIZE];
54     printf("Enter an infix expression: ");
55     fgets(infix, MAX_SIZE, stdin);
56     infixToPostfix(infix);
57     printf("Postfix expression: %s\n", infix);
58     return 0;
59 }
```

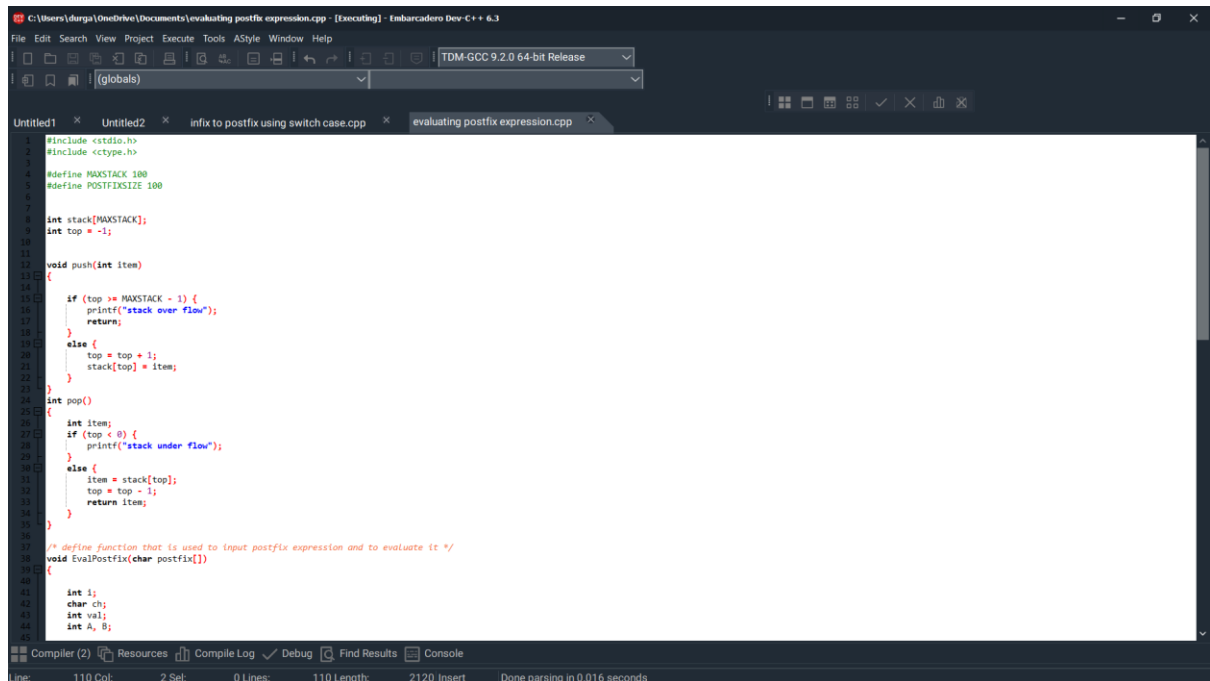
Compiler Resources Compile Log Debug Find Results Console

Line: 1 Col: 1 Sel: 0 Lines: 103 Length: 2559 Insert Done parsing in 0.203 seconds

```
C:\Users\durga\OneDrive\Dev
Enter an infix expression: ((a+b)*c)
Postfix expression: ab+c*

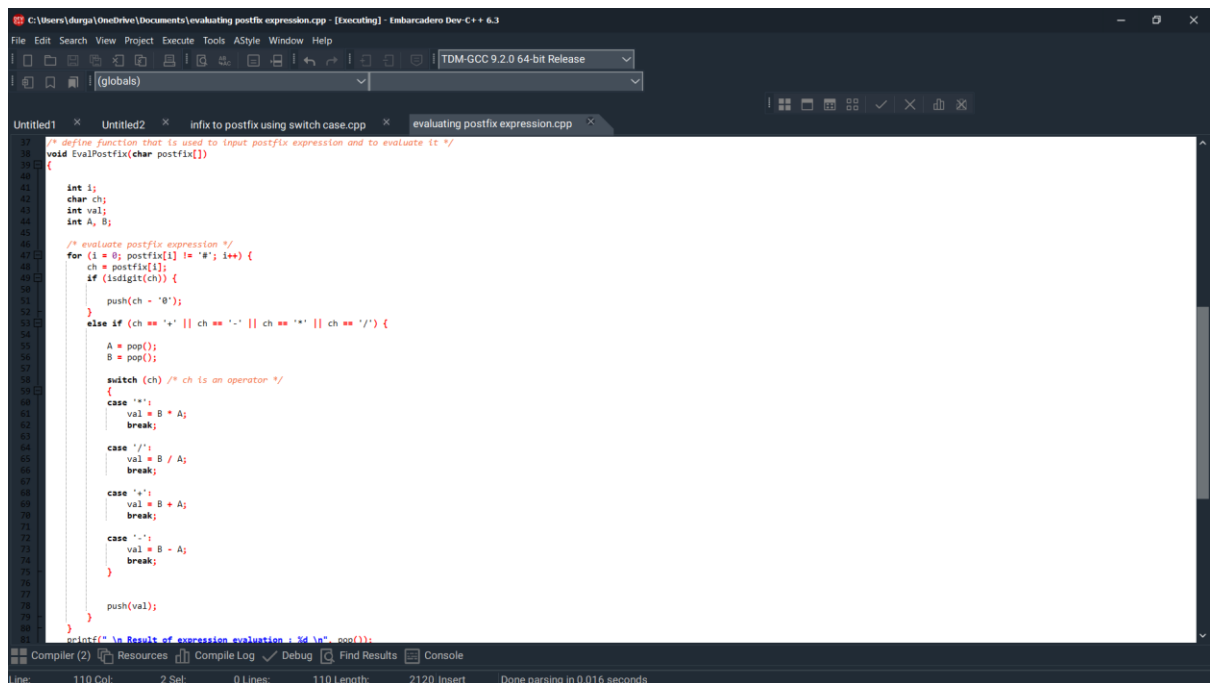
-----
Process exited after 18.43 seconds with return value 0
Press any key to continue . . .
```

2)evaluating postfix equation



```
C:\Users\durga\OneDrive\Documents\evaluating postfix expression.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Untitled1 x Untitled2 x infix to postfix using switch case.cpp x evaluating postfix expression.cpp x
1 #include <stdio.h>
2 #include <ctype.h>
3
4 #define MAXSTACK 100
5 #define POSTFIXSIZE 100
6
7
8 int stack[MAXSTACK];
9 int top = -1;
10
11 void push(int item)
12 {
13     if (top >= MAXSTACK - 1) {
14         printf("stack over flow");
15         return;
16     }
17     else {
18         top = top + 1;
19         stack[top] = item;
20     }
21 }
22
23 int pop()
24 {
25     int item;
26     if (top < 0) {
27         printf("stack under flow");
28     }
29     else {
30         item = stack[top];
31         top = top - 1;
32         return item;
33     }
34 }
35
36 /* define function that is used to input postfix expression and to evaluate it */
37 void EvalPostfix(char postfix[])
38 {
39     int i;
40     char ch;
41     int val;
42     int A, B;
43 }
```

Compiler (2) Resources Compile Log Debug Find Results Console
Line: 110 Col: 2 Sel: 0 Lines: 110 Length: 2120 Insert Done parsing in 0.016 seconds



```
C:\Users\durga\OneDrive\Documents\evaluating postfix expression.cpp - [Executing] - Embarcadero Dev-C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)
Untitled1 x Untitled2 x infix to postfix using switch case.cpp x evaluating postfix expression.cpp x
37 /* define function that is used to input postfix expression and to evaluate it */
38 void EvalPostfix(char postfix[])
39 {
40     int i;
41     char ch;
42     int val;
43     int A, B;
44
45     /* evaluate postfix expression */
46     for (i = 0; postfix[i] != '\0'; i++) {
47         ch = postfix[i];
48         if (isdigit(ch)) {
49             push(ch - '0');
50         }
51         else if (ch == '+' || ch == '-' || ch == '*' || ch == '/') {
52             A = pop();
53             B = pop();
54             switch (ch) /* ch is an operator */
55             {
56                 case '+':
57                     val = B + A;
58                     break;
59                 case '/':
60                     val = B / A;
61                     break;
62                 case '*':
63                     val = B * A;
64                     break;
65                 case '-':
66                     val = B - A;
67                     break;
68             }
69             push(val);
70         }
71     }
72     printf("Result of expression evaluation is %d\n", pop());
73 }
```

Compiler (2) Resources Compile Log Debug Find Results Console
Line: 110 Col: 2 Sel: 0 Lines: 110 Length: 2120 Insert Done parsing in 0.016 seconds

```
C:\Users\durga\OneDrive\Documents\evaluating postfix expression.cpp - [Executing] - Embarcadero Dev C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)

Untitled1 x Untitled2 x infix to postfix using switch case.cpp x evaluating postfix expression.cpp x

67     case '+':
68         val = B + A;
69         break;
70
71     case '-':
72         val = B - A;
73         break;
74     }
75
76     push(val);
77 }
78
79 printf("\n Result of expression evaluation : %d \n", pop());
80
81
82
83
84 int main()
85 {
86     int i;
87
88     /* declare character array to store postfix expression */
89     char postfix[POSTFIXSIZE];
90     printf("ASSUMPTION: There are only four operators(*, /, +, -) in an expression and operand is single digit only.\n");
91     printf("\nEnter postfix expression,\npress right parenthesis '#' for end expression : ");
92
93
94
95     for (i = 0; i <= POSTFIXSIZE - 1; i++) {
96         scanf("%c", &postfix[i]);
97
98         if (postfix[i] == '#')
99         {
100             break;
101         }
102     }
103
104
105     EvalPostfix(postfix);
106
107     return 0;
108 }

Compiler (2) Resources Compile Log Debug Find Results Console
Line: 110 Col: 2 Sel: 0 Lines: 110 Length: 2120 Insert Done parsing in 0.016 seconds
```

```
C:\Users\durga\OneDrive\Documents\evaluating postfix expression.cpp - [Executing] - Embarcadero Dev C++ 6.3
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 9.2.0 64-bit Release
(globals)

Untitled1 x C:\Users\durga\OneDrive\Documents\evaluating postfix expression.cpp x + v x
+ v x

67     case '+':
68         val = B + A;
69         break;
70
71     case '-':
72         val = B - A;
73         break;
74     }
75
76     push(val);
77 }
78
79 printf("\n Result of expression evaluation : %d \n", pop());
80
81
82
83
84 int main()
85 {
86     int i;
87
88     /* declare character array to store postfix expression */
89     char postfix[POSTFIXSIZE];
90     printf("ASSUMPTION: There are only four operators(*, /, +, -) in an expression and operand is single digit only.\n");
91     printf("\nEnter postfix expression,\npress right parenthesis '#' for end expression : ");
92
93
94
95     for (i = 0; i <= POSTFIXSIZE - 1; i++) {
96         scanf("%c", &postfix[i]);
97
98         if (postfix[i] == '#')
99         {
100             break;
101         }
102     }
103
104
105     EvalPostfix(postfix);
106
107     return 0;
108 }

Compiler (2) Resources Compile Log Debug Find Results Console
Line: 110 Col: 2 Sel: 0 Lines: 110 Length: 2120 Insert Done parsing in 0.016 seconds
```

```
ASSUMPTION: There are only four operators(*, /, +, -) in an expression and operand is single digit only.
Enter postfix expression,
press right parenthesis '#' for end expression : 54+#

Result of expression evaluation : 9

-----
Process exited after 24.38 seconds with return value 0
Press any key to continue . . .
```