

EXERCISE - 4**CODE TO CONVERT INFIX EXPRESSION TO POSTFIX EXPRESSION**

```
#include<stdio.h>
char stack[20];
int top = -1;
void push(char x)
{
    stack[++top] = x;
}

char pop()
{
    if(top == -1)
        return -1;
    else
        return stack[top--];
}

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

main()
{
    char exp[20];
    char *e, x;
    printf("Enter the expression :: ");
    scanf("%s",exp);
    e = exp;
    while(*e != '\0')
    {
        if(isalnum(*e))
            printf("%c",*e);
        else if(*e == '(')
            push(*e);
        else if(*e == ')')
```

```

    {
        while((x = pop()) != '(')
            printf("%c", x);
    }
    else
    {
        while(priority(stack[top]) >= priority(*e))
            printf("%c", pop());
        push(*e);
    }
    e++;
}
while(top != -1)
{
    printf("%c", pop());
}
}

```

The screenshot shows the Code::Blocks IDE with a C program open in the editor. The program implements an infix to postfix conversion algorithm using a stack. The code is as follows:

```

1  #include<stdio.h>
2  char stack[20];
3  int top = -1;
4  void push(char x)
5  {
6      stack[++top] = x;
7  }
8
9  char pop()
10 {
11     if(top == -1)
12         return -1;
13     else
14         return stack[top--];
15 }
16
17 int priority(char x)
18 {
19     if(x == '(')
20         return 0;
21     if(x == '+' || x == '-')
22         return 1;
23     if(x == '*' || x == '/')
24         return 2;
25 }
26
27 main()
28 {
29     char exp[20];
30     char *e, x;
31     printf("Enter the expression :: ");

```

The IDE interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help), a toolbar, and a project manager on the left. The status bar at the bottom shows the file path, line and column numbers, and the current time and date.

```
27 main()
28 {
29     char exp[20];
30     char *e, x;
31     printf("Enter the expression :: ");
32     scanf("%s", exp);
33     e = exp;
34     while(*e != '\0')
35     {
36         if(isalnum(*e))
37             printf("%c", *e);
38         else if(*e == '(')
39             push(*e);
40         else if(*e == ')')
41         {
42             while((x = pop()) != '(')
43                 printf("%c", x);
44         }
45         else
46         {
47             while(priority(stack[top]) >= priority(*e))
48                 printf("%c", pop());
49             push(*e);
50         }
51         e++;
52     }
53     while(top != -1)
54     {
55         printf("%c", pop());
56     }
57 }
```

OUTPUT :

```
gcc version 4.6.3
main.c:27:1: warning: return type defaults to 'int' [-Wimplicit-int]
main()
^~~~~
main.c: In function 'main':
main.c:36:12: warning: implicit declaration of function 'isalnum' [-Wimplicit-function-declaration]
    if(isalnum(*e))
       ^~~~~~
Enter the expression :: (a+b)*(c-d)/(e+f)
ab+cd-*ef+/> 
```

CODE TO EVALUATE POSTFIX EXPRESSION

```
#include<stdio.h>
int stack[20];
int top = -1;

void push(int x)
{
    stack[++top] = x;
}

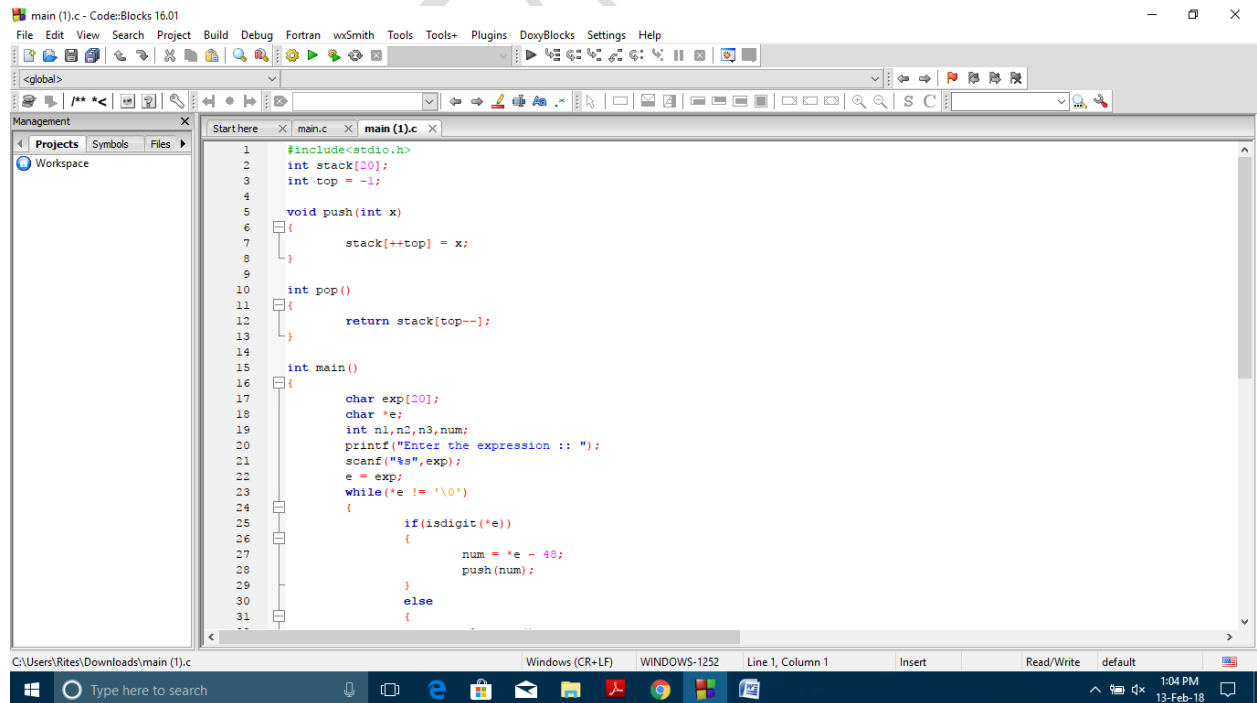
int pop()
{
    return stack[top--];
}

int main()
{
    char exp[20];
    char *e;
    int n1,n2,n3,num;
    printf("Enter the expression :: ");
    scanf("%s",exp);
    e = exp;
    while(*e != '\0')
    {
        if(isdigit(*e))
        {
            num = *e - 48;
            push(num);
        }
        else
        {
            n1 = pop();
            n2 = pop();
            switch(*e)
            {
                case '+':
                {
                    n3 = n1 + n2;
                    break;
                }
                case '-':
                {
                    n3 = n2 - n1;
                    break;
                }
            }
        }
    }
}
```

```

    }
    case '*':
    {
        n3 = n1 * n2;
        break;
    }
    case '/':
    {
        n3 = n2 / n1;
        break;
    }
    }
    push(n3);
}
e++;
}
printf("\nThe result of expression %s =
%d\n\n",exp,pop());
return 0;
}

```



The screenshot shows the Code::Blocks IDE with a C program for evaluating postfix expressions. The code is as follows:

```

1  #include<stdio.h>
2  int stack[20];
3  int top = -1;
4
5  void push(int x)
6  {
7      stack[++top] = x;
8  }
9
10 int pop()
11 {
12     return stack[top--];
13 }
14
15 int main()
16 {
17     char exp[20];
18     char *e;
19     int n1,n2,n3,num;
20     printf("Enter the expression :: ");
21     scanf("%s",exp);
22     e = exp;
23     while(*e != '\0')
24     {
25         if(isdigit(*e))
26         {
27             num = *e - 48;
28             push(num);
29         }
30         else
31         {

```

The IDE interface includes a menu bar (File, Edit, View, Search, Project, Build, Debug, Fortran, wxSmith, Tools, Tools+, Plugins, DoxyBlocks, Settings, Help), a toolbar, and a sidebar with 'Projects' and 'Workspace' tabs. The main editor window displays the code, and the status bar at the bottom shows the file path, line/column information, and system clock.

main (1).c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management

Projects Symbols Files

Workspace

```
15 int main()
16 {
17     char exp[20];
18     char *e;
19     int n1,n2,n3,num;
20     printf("Enter the expression :: ");
21     scanf("%s",exp);
22     e = exp;
23     while(*e != '\0')
24     {
25         if(isdigit(*e))
26         {
27             num = *e - 48;
28             push(num);
29         }
30         else
31         {
32             n1 = pop();
33             n2 = pop();
34             switch(*e)
35             {
36                 case '+':
37                     n3 = n1 + n2;
38                     break;
39                 case '-':
40                     n3 = n2 - n1;
41                     break;
42             }
43         }
44     }
45 }
```

C:\Users\Rites\Downloads\main (1).c

Windows (CR+LF) WINDOWS-1252 Line 1, Column 1 Insert Read/Write default 1:05 PM 13-Feb-18

main (1).c - Code::Blocks 16.01

File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help

<global>

Management

Projects Symbols Files

Workspace

```
34 switch(*e)
35 {
36     case '+':
37         n3 = n1 + n2;
38         break;
39     case '-':
40         n3 = n2 - n1;
41         break;
42     case '*':
43         n3 = n1 * n2;
44         break;
45     case '/':
46         n3 = n2 / n1;
47         break;
48 }
49 push(n3);
50 e++;
51 }
52 printf("\nThe result of expression %s = %d\n",exp,pop());
53 return 0;
54 }
```

C:\Users\Rites\Downloads\main (1).c

Windows (CR+LF) WINDOWS-1252 Line 1, Column 1 Insert Read/Write default 1:05 PM 13-Feb-18

OUTPUT :

```
gcc version 4.6.3
```

```
>
```

```
main.c: In function 'main':
```

```
main.c:25:20: warning: implicit declaration of function  
'isdigit' [-Wimplicit-function-declaration]
```

```
    if(isdigit(*e))
```

```
        ^~~~~~
```

```
Enter the expression :: (300+23)*(43-21)/(84+7)
```

```
The result of expression (300+23)*(43-21)/(84+7) = 12
```

```
>
```