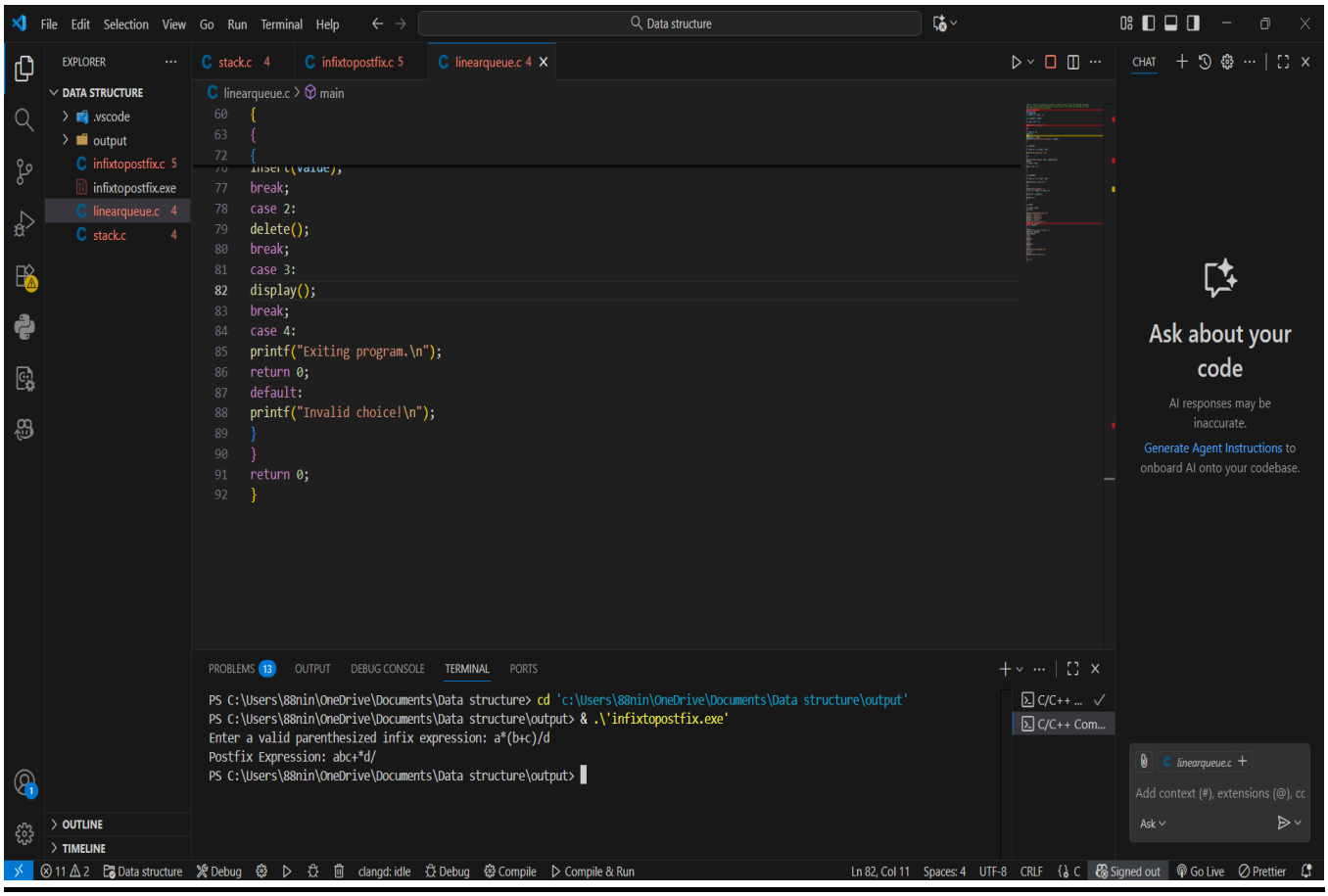
/*WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide)*/

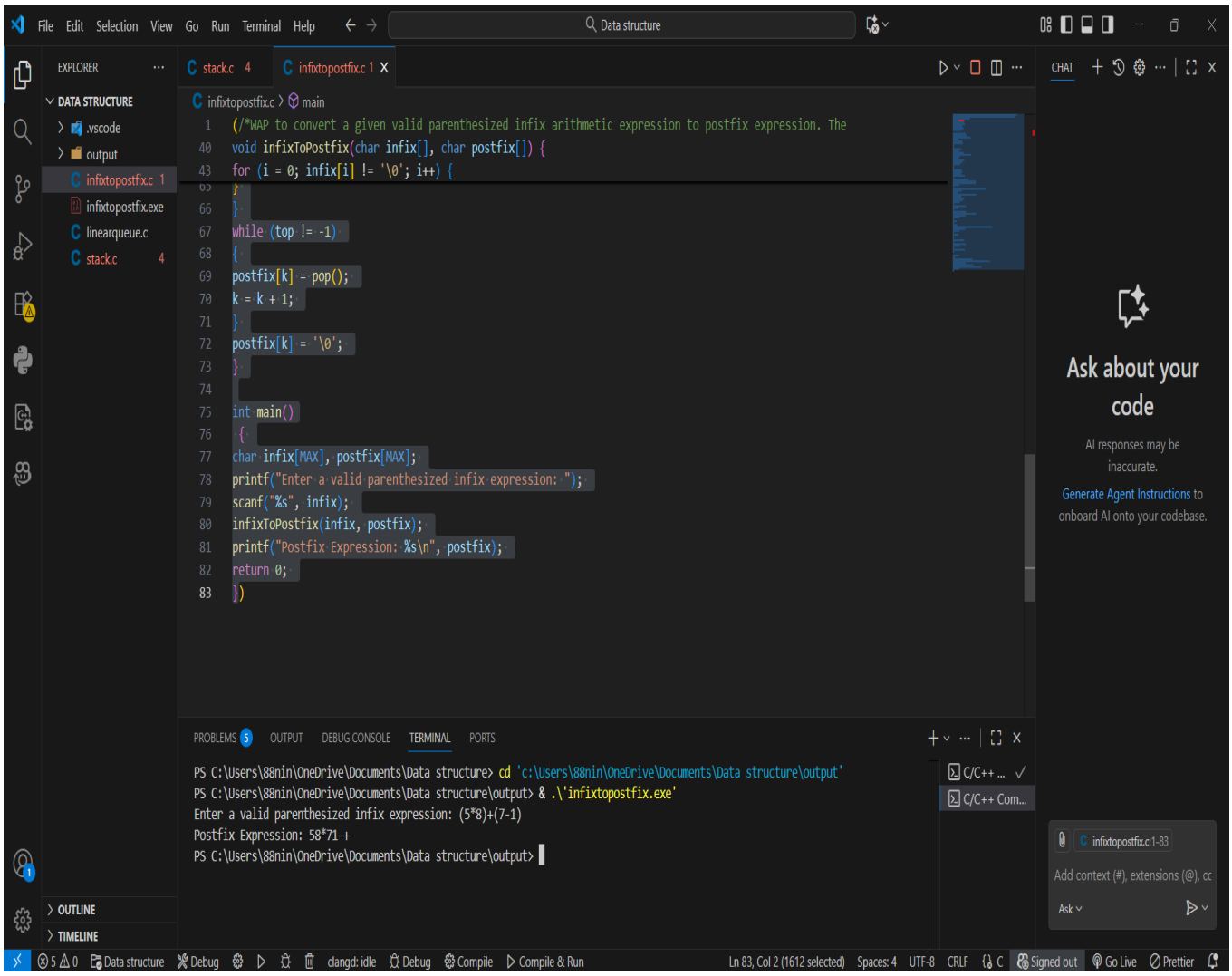
```
#include <stdio.h>

#include <ctype.h>

#include <string.h>

#define MAX 100

char stack[MAX];

int top = -1;


void push(char c) {

if (top == MAX - 1) {

printf("Stack Overflow\n");

} else {

top = top + 1;

stack[top] = c;

}

}

char pop() {

char val;

if (top == -1) {

printf("Stack Underflow\n");

return -1;

} else {

val = stack[top];

top = top - 1;

return val;

}

}
```

```
char peek() {
if (top == -1)
return '\0';
return stack[top];
}
int precedence(char c) {
if (c == '+' || c == '-') return 1;
if (c == '*' || c == '/') return 2;
return 0;
}
void infixToPostfix(char infix[], char postfix[]) {
int i, k = 0;
char c;
for (i = 0; infix[i] != '\0'; i++) {
c = infix[i];
if (isalnum(c)) {
postfix[k] = c;
k = k + 1;
}
else if (c == '(') {
push(c);
}
else if (c == ')') {
while (top != -1 && peek() != '(') {
postfix[k] = pop();
k = k + 1;
}
pop();
```

```
}
else {
while (top != -1 && precedence(peek()) >= precedence(c)) {
postfix[k] = pop();
k = k + 1;
}
push(c);
}
}
while (top != -1)
{
postfix[k] = pop();
k = k + 1;
}
postfix[k] = '\0';
}

int main()
 {
char infix[MAX], postfix[MAX];
printf("Enter a valid parenthesized infix expression: ");
scanf("%s", infix);
infixToPostfix(infix, postfix);
printf("Postfix Expression: %s\n", postfix);
return 0;
}
```

Output:-


File Edit Selection View Go Run Terminal Help
Data structure
EXPLORER
DATA STRUCTURE
.vscode
output
infixtopostfix.c 5
infixtopostfix.exe
linearqueue.c 4
stack.c 4
stack.c 4
infixtopostfix.c 5
linearqueue.c 4
linearqueue.c > main
60 {
63 {
72 {
77 break;
78 case 2:
79 delete();
80 break;
81 case 3:
82 display();
83 break;
84 case 4:
85 printf("Exiting program.\n");
86 return 0;
87 default:
88 printf("Invalid choice!\n");
89 }
90 }
91 return 0;
92 }
CHAT
Ask about your code
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
PROBLEMS 13
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\88nin\OneDrive\Documents\Data structure> cd 'c:\Users\88nin\OneDrive\Documents\Data structure\output'
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\'infixtopostfix.exe'
Enter a valid parenthesized infix expression: a*(b+c)/d
Postfix Expression: abc+*d/
PS C:\Users\88nin\OneDrive\Documents\Data structure\output>
C/C++ ...
C/C++ Com...
linearqueue.c
Add context (#), extensions (@), cc
Ask
OUTLINE
TIMELINE
Data structure
Debug
clangd: idle
Debug
Compile
Compile & Run
Ln 82, Col 11
Spaces: 4
UTF-8
CRLF
C
Signed out
Go Live
Prettier


File Edit Selection View Go Run Terminal Help
Data structure
EXPLORER
DATA STRUCTURE
.vscode
output
infixtopostfix.c 1
infixtopostfix.exe
linearqueue.c
stack.c 4
stack.c 4
infixtopostfix.c 1
infixtopostfix.c > main
1 (/*WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The
40 void infixToPostfix(char infix[], char postfix[]) {
43 for (i = 0; infix[i] != '\0'; i++) {
66 }
67 while (top != -1)
68 {
69 postfix[k] = pop();
70 k = k + 1;
71 }
72 postfix[k] = '\0';
73 }
74
75 int main()
76 {
77 char infix[MAX], postfix[MAX];
78 printf("Enter a valid parenthesized infix expression: ");
79 scanf("%s", infix);
80 infixToPostfix(infix, postfix);
81 printf("Postfix Expression: %s\n", postfix);
82 return 0;
83 }
PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\88nin\OneDrive\Documents\Data structure> cd 'c:\Users\88nin\OneDrive\Documents\Data structure\output'
PS C:\Users\88nin\OneDrive\Documents\Data structure\output> & .\'infixtopostfix.exe'
Enter a valid parenthesized infix expression: (5*8)+(7-1)
Postfix Expression: 58*71-+
PS C:\Users\88nin\OneDrive\Documents\Data structure\output>
C/C++ ...
C/C++ Com...
CHAT
Ask about your code
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
infixtopostfix.c:1-83
Add context (#), extensions (@), cc
Ask
OUTLINE
TIMELINE
Data structure
Debug
clangd: idle
Debug
Compile
Compile & Run
Ln 83, Col 2 (1612 selected)
Spaces: 4
UTF-8
CRLF
C
Signed out
Go Live
Prettier