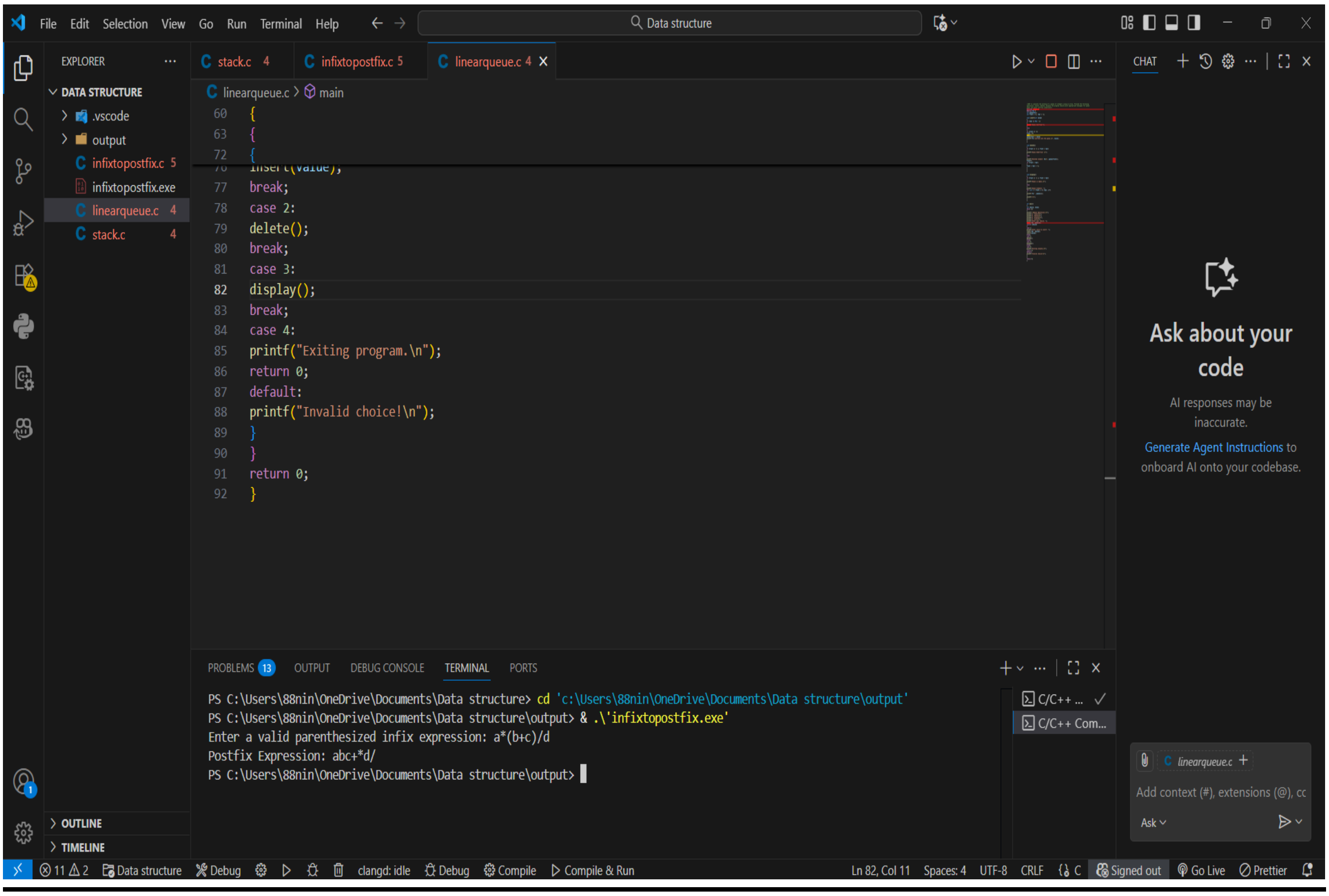
```
/*WAP to convert a given valid parenthesized infix arithmetic expression to postfix
expression. The expression consists of single character operands and the binary operators +
(plus), - (minus), * (multiply) and / (divide)*/
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#define MAX 100
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();
```

```c
}
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:-



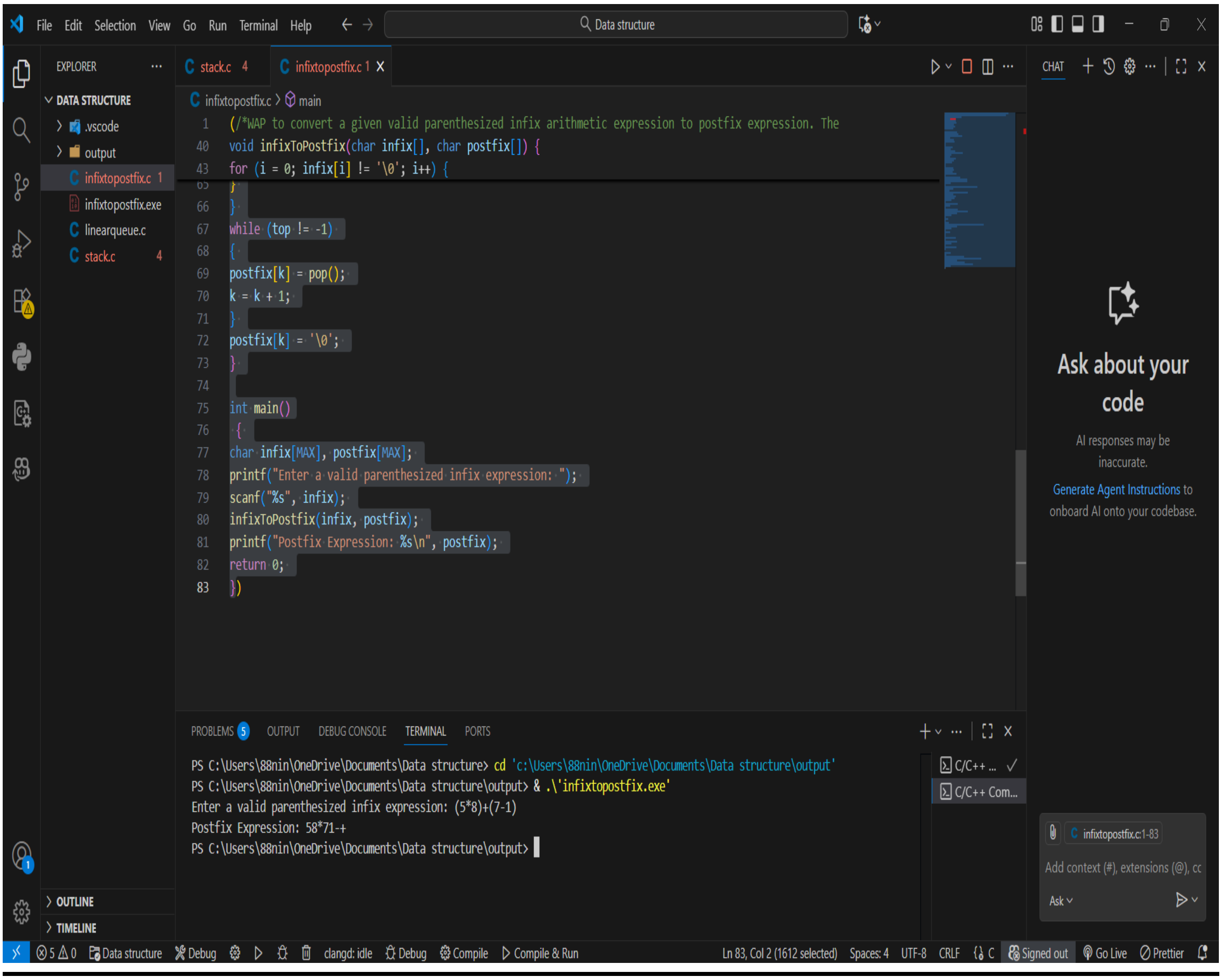
```
{
{
{
insert(value);
break;
case 2:
delete();
break;
case 3:
display();
break;
case 4:
printf("Exiting program.\n");
return 0;
default:
printf("Invalid choice!\n");
}
}
return 0;
}
```

```
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>
```

Ask about your code

AI responses may be inaccurate.

Generate Agent Instructions to onboard AI onto your codebase.



```c
(/*WAP to convert a given valid parenthesized infix arithmetic expression to postfix expression. The
void infixToPostfix(char infix[], char postfix[]) {
for (i = 0; infix[i] != '\0'; i++) {
}
}
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;
})
```

```
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>
```