

Design a lexical Analyzer to find the number of whitespaces and newline characters.

PGM:

```
#include <stdio.h>
```

```
#define MAX_LENGTH 1000
```

```
void countWhitespacesAndNewlines(const char *input, int *whitespaceCount, int *newlineCount) {
```

```
    int i = 0;
```

```
    while (input[i] != '\0') {
```

```
        if (input[i] == ' ' || input[i] == '\t') {
```

```
            (*whitespaceCount)++;
```

```
        } else if (input[i] == '\n') {
```

```
            (*newlineCount)++;
```

```
        }
```

```
        i++;
```

```
    }
```

```
}
```

```
int main() {
```

```
    char input[MAX_LENGTH];
```

```
    int whitespaceCount = 0;
```

```
    int newlineCount = 0;
```

```
    printf("Enter text (Press Enter twice to finish):\n");
```

```
    // Read multiple lines of input until a blank line is entered
```

```
    while (fgets(input, MAX_LENGTH, stdin) && input[0] != '\n') {
```

```
        countWhitespacesAndNewlines(input, &whitespaceCount, &newlineCount);
```

```
}
```




```
printf("Number of whitespace characters: %d\n", whitespaceCount);
```

```
printf("Number of newline characters: %d\n", newlineCount);
```

```
return 0;
```

```
}
```

OUTPUT:

main.c	   Share	Run	Output
11	(*newlineCount)++;		Enter text (Press Enter twice to finish):
12	}		4th experiment ON-progress
13	i++;		
14	}		Number of whitespace characters: 2
15	}		Number of newline characters: 1
16			
17	int main() {		
18	char input[MAX_LENGTH];		
19	int whitespaceCount = 0;		
20	int newlineCount = 0;		
21			
22	printf("Enter text (Press Enter twice to finish):\n");		
23			
24	// Read multiple lines of input until a blank line is entered		
25	while (fgets(input, MAX_LENGTH, stdin) && input[0] != '\n') {		
26	countWhitespacesAndNewlines(input, &whitespaceCount,		
	&newlineCount);		
27	}		
28			
29	printf("Number of whitespace characters: %d\n", whitespaceCount		
);		
30	printf("Number of newline characters: %d\n", newlineCount);		
31			
32	return 0;		
33	}		

=== Code Execution Successful ===