

CSA 1447- COMPILER DESIGN FOR SYNTAX SMITH

PRACTICAL PROGRAMS 15-21

EXPERIMENT -15

PROGRAM:

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

```
#include <ctype.h>
```

```
int main() {
```

```
    char str[] = "#include <stdio.h>\nint main() {\n    int a = 10;\n    printf(\"Hello World\");\n    return 0;\n}\n";
```

```
    int char_count = 0, word_count = 0, line_count = 1;
```

```
    int in_word = 0;
```

```
    printf("Input Source Program:\n%s\n\n", str);
```

```
    for (int i = 0; str[i] != '\0'; i++) {
```

```
        char_count++;
```

```
        if (str[i] == '\n') line_count++;
```

```
        if (isspace(str[i])) {
```

```
            in_word = 0;
```

```
        } else if (!in_word) {
```

```
            in_word = 1;
```

```
            word_count++;
```

```
        }
```

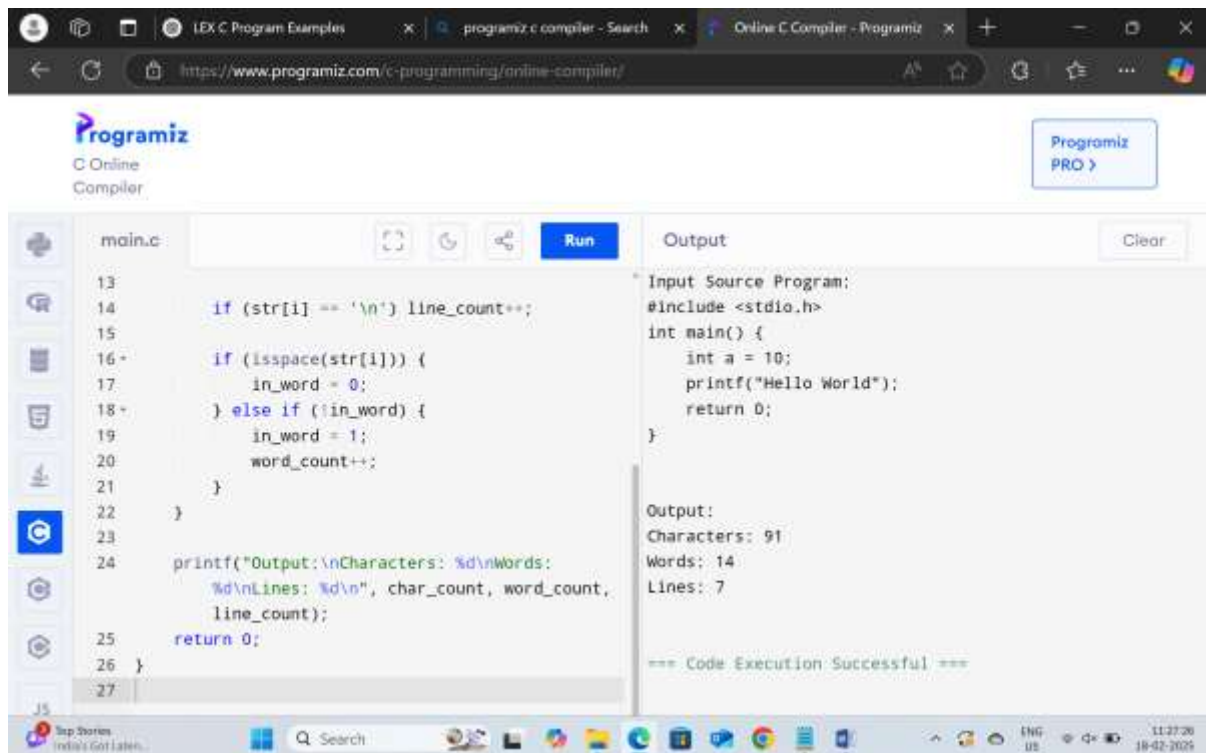
```
    }
```

```
    printf("Output:\nCharacters: %d\nWords: %d\nLines: %d\n", char_count, word_count, line_count);
```

```
    return 0;
```

```
}
```

OUTPUT:



The screenshot shows the Programiz Online C Compiler interface. The code editor on the left contains a C program that counts characters, words, and lines in a string. The output window on the right displays the input source program and the execution results.

```
main.c
13
14     if (str[i] == '\n') line_count++;
15
16     if (isspace(str[i])) {
17         in_word = 0;
18     } else if (!in_word) {
19         in_word = 1;
20         word_count++;
21     }
22 }
23
24 printf("Output:\nCharacters: %d\nWords: %d\nLines: %d", char_count, word_count, line_count);
25 return 0;
26 }
27
```

Output

Input Source Program:

```
#include <stdio.h>
int main() {
    int a = 10;
    printf("Hello World");
    return 0;
}
```

Output:

Characters: 91
Words: 14
Lines: 7

=== Code Execution Successful ===

EXPERIMENT-16

PROGRAM:

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

```
#include <ctype.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char str[] = "#define PI 3.14\n#include<stdio.h>\nint main() {\n    int a = 10, b = 20;\n    printf("%d", a + b);\n    return 0;\n}\n";
```

```
    printf("Input Source Program:\n%s\n", str);
```

```
    printf("Output:\n");
```

```
    for (int i = 0; str[i] != '\0'; i++) {
```

```
        if (isdigit(str[i])) {
```

```
            while (isdigit(str[i]) || str[i] == '.') {
```

```
                printf("%c", str[i]);
```

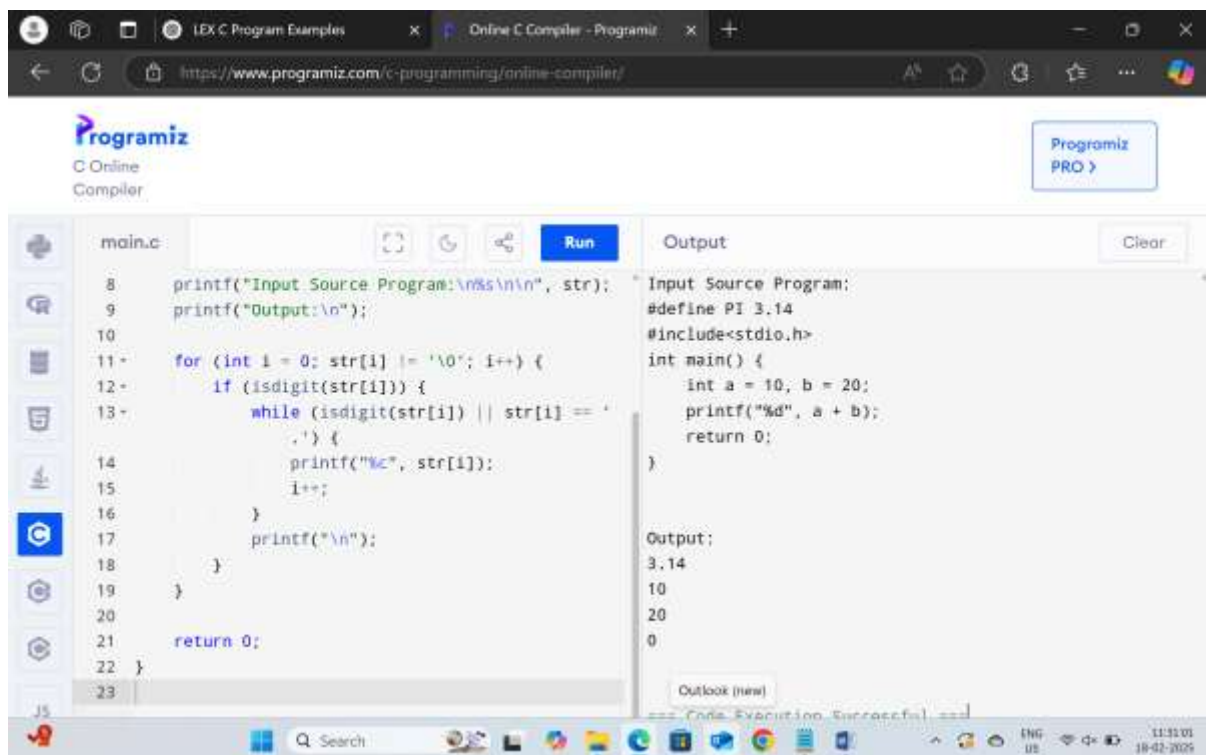
```

        i++;
    }
    printf("\n");
}

return 0;
}

```

OUTPUT:



EXPERIMENT-17

PROGRAM:

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

```
#include <string.h>
```

```
int main() {
```

```
    char str[] = "#define PI 3.14\n#include<stdio.h>\n#include<conio.h>\nint main() { return 0; }\n";
```

```
    int macro_count = 0, header_count = 0;
```

```
printf("Input Source Program:\n%s\n\n", str);
```

```
char *line = strtok(str, "\n");
```

```
while (line) {
```

```
    if (strncmp(line, "#define", 7) == 0) macro_count++;
```

```
    if (strncmp(line, "#include", 8) == 0) header_count++;
```

```
    line = strtok(NULL, "\n");
```

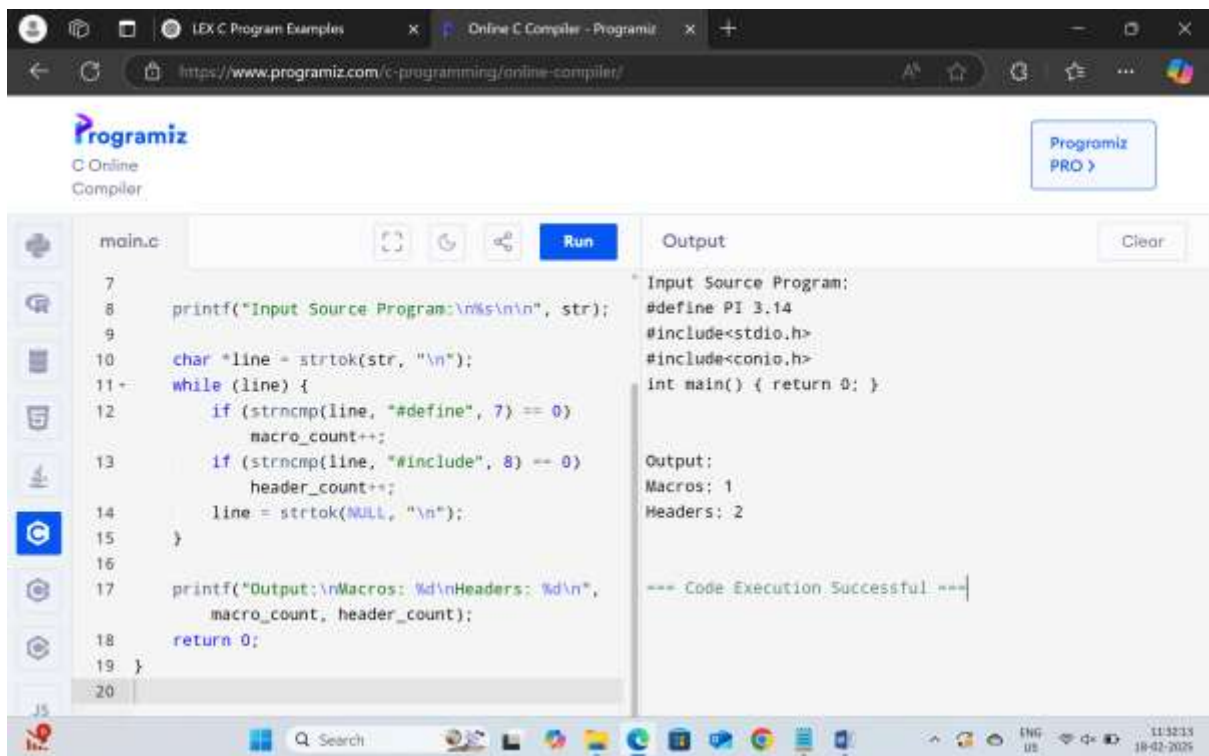
```
}
```

```
printf("Output:\nMacros: %d\nHeaders: %d\n", macro_count, header_count);
```

```
return 0;
```

```
}
```

OUTPUT:



The screenshot displays the Programiz Online C Compiler interface. The code editor on the left contains a C program that takes an input source program and counts the number of macros and headers. The output window on the right shows the results of the compilation and execution.

```
main.c
7
8 printf("Input Source Program:\n%s\n\n", str);
9
10 char *line = strtok(str, "\n");
11 while (line) {
12     if (strncmp(line, "#define", 7) == 0) macro_count++;
13     if (strncmp(line, "#include", 8) == 0) header_count++;
14     line = strtok(NULL, "\n");
15 }
16
17 printf("Output:\nMacros: %d\nHeaders: %d\n", macro_count, header_count);
18 return 0;
19 }
20
```

Output:

```
Input Source Program:
#define PI 3.14
#include<stdio.h>
#include<conio.h>
int main() { return 0; }

Output:
Macros: 1
Headers: 2

--- Code Execution Successful ---
```

EXPERIMENT 18

PROGRAM:

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

```
#include <string.h>
```

```

int main() {

    char str[] = "#define PI 3.14\n#include<stdio.h>\n#include<conio.h>\nint main() { return 0; }\n";

    int macro_count = 0, header_count = 0;

    printf("Input Source Program:\n%s\n\n", str);

    char *line = strtok(str, "\n");
    while (line) {
        if (strncmp(line, "#define", 7) == 0) macro_count++;
        if (strncmp(line, "#include", 8) == 0) header_count++;
        line = strtok(NULL, "\n");
    }

    printf("Output:\nMacros: %d\nHeaders: %d\n", macro_count, header_count);
    return 0;
}

```

OUTPUT:

The screenshot shows the Programiz Online C Compiler interface. The code editor on the left contains the C program, and the output window on the right displays the results of the compilation and execution.

Code Editor (main.c):

```

7
8     printf("Input Source Program:\n%s\n\n", str);
9
10    char *line = strtok(str, "\n");
11    while (line) {
12        if (strncmp(line, "#define", 7) == 0)
13            macro_count++;
14        if (strncmp(line, "#include", 8) == 0)
15            header_count++;
16        line = strtok(NULL, "\n");
17    }
18
19    printf("Output:\nMacros: %d\nHeaders: %d\n",
20          macro_count, header_count);
21    return 0;

```

Output Window:

```

Input Source Program:
#define PI 3.14
#include<stdio.h>
#include<conio.h>
int main() { return 0; }

Output:
Macros: 1
Headers: 2

--- Code Execution Successful ---

```

EXPERIMENT-19

PROGRAM:

```
#include <stdio.h>

int main() {
    char str[] = "<html>\n<body>\n<h1>Title</h1>\n<p>Paragraph</p>\n</body>\n</html>\n";

    printf("Input HTML File:\n%s\n\n", str);
    printf("Output:\n");

    int inside_tag = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        if (str[i] == '<') {
            inside_tag = 1;
            printf("Tag: ");
        }

        if (inside_tag) {
            printf("%c", str[i]);
        }

        if (str[i] == '>') {
            inside_tag = 0;
            printf("\n");
        }
    }

    return 0;
}
```

OUTPUT:

The screenshot shows a web browser window with the URL <https://www.programiz.com/c-programming/online-compiler/>. The page title is "Programiz C Online Compiler". The code editor contains the following C program:

```
main.c
4 char str[] = "<html>\n<body>\n<h1>Title</h1>\n<p>Paragraph</p>\n>\n</body>\n</html>\n";
5
6 printf("Input HTML File:\n%s\n", str);
7 printf("Output:\n");
8
9 int inside_tag = 0;
10 for (int i = 0; str[i] != '\0'; i++) {
11     if (str[i] == '<') {
12         inside_tag = 1;
13         printf("Tag: ");
14     }
15
16     if (inside_tag) {
17         printf("%c", str[i]);
18     }
19
20     if (str[i] == '>') {
21         inside_tag = 0;
22         printf("\n");
23     }
24 }
25
26 return 0;
```

The output window shows the following results:

```
Input HTML File:
<html>
<body>
<h1>Title</h1>
<p>Paragraph</p>
</body>
</html>

Output:
Tag: <html>
Tag: <body>
Tag: <h1>
Tag: </h1>
Tag: <p>
Tag: </p>
Tag: </body>
Tag: </html>

=== Code Execution Successful ===
```

EXPERIMENT-20

PROGRAM:

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

```
int main() {
```

```
    char str[] = "#define PI 3.14\n#include<stdio.h>\nint main() {\n    int a = 10;\n    printf(\"Hello\");\n    return 0;\n}\n";
```

```
    int line_num = 1;
```

```
    printf("Input C Program:\n%s\n", str);
```

```
    printf("Output:\n");
```

```
    printf("%d: ", line_num++);
```

```
    for (int i = 0; str[i] != '\0'; i++) {
```

```
        printf("%c", str[i]);
```

```
        if (str[i] == '\n' && str[i + 1] != '\0') {
```

```

        printf("%d: ", line_num++);
    }
}

return 0;
}

```

OUTPUT:

The screenshot shows the Programiz Online C Compiler interface. The code editor on the left contains a C program that reads a string from the user, which is a small C program itself. The program prints the string and then prints the line numbers of the input string. The output on the right shows the execution results.

```

main.c
1 #include <stdio.h>
2
3 int main() {
4     char str[] = "#include <stdio.h>\nint main()
5     {\n    int a = 10;\n    printf(\"Hello\\n\");\n    return 0
6     }\n}";
7     int line_num = 1;
8
9     printf("Input C Program:\\n\\n", str);
10    printf("Output:\\n");
11
12    printf("%d: ", line_num++);
13    for (int i = 0; str[i] != '\\0'; i++) {
14        printf("%c", str[i]);
15        if (str[i] == '\\n' && str[i + 1] != '\\0') {
16            printf("%d: ", line_num++);
17        }
18    }
19    return 0;
20 }
21

```

Output

```

Input C Program:
#include <stdio.h>
int main() {
    int a = 10;
    printf("Hello");
    return 0;
}

Output:
1: #include <stdio.h>
2: #include <stdio.h>
3: int main() {
4:     int a = 10;
5:     printf("Hello");
6:     return 0;
7: }

=== Code Execution Successful ===

```

EXPERIMENT-21

PROGRAM:

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

```
#include <ctype.h>
```

```
int main() {
```

```
    // Sample Input: A small C program stored in a string
```

```
    char input[] =
```

```
        "#include <stdio.h>\n"
```

```
        "int main() {\n"
```



```

"    int a = 10, b = 20;\n"
"    printf(\"Hello, World!\");\n"
"    return 0;\n"
"}\n";

int char_count = 0, word_count = 0, line_count = 1;
int in_word = 0;

printf("Input Source Program:\n%s\n\n", input);

for (int i = 0; input[i] != '\0'; i++) {
    char_count++;

    if (input[i] == '\n')
        line_count++;

    if (isspace(input[i])) {
        in_word = 0; // End of a word
    } else if (!in_word) {
        in_word = 1;
        word_count++; // Start of a new word
    }
}

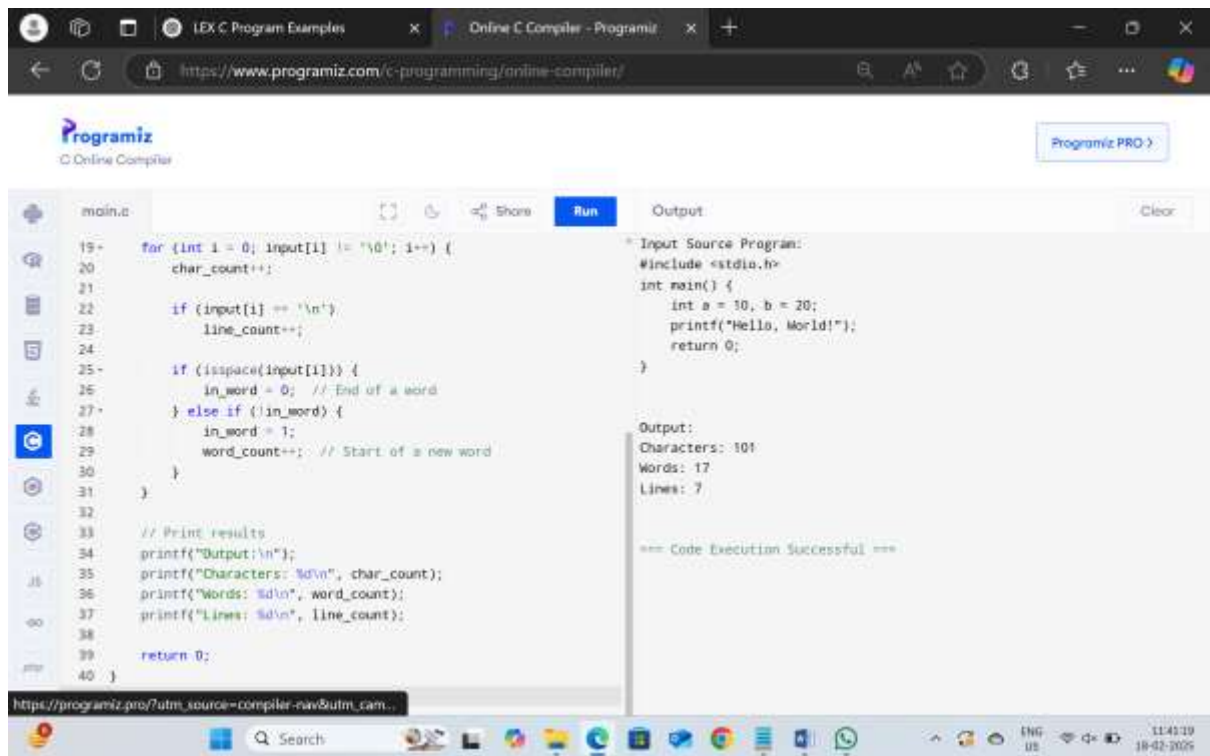
// Print results
printf("Output:\n");
printf("Characters: %d\n", char_count);
printf("Words: %d\n", word_count);
printf("Lines: %d\n", line_count);

return 0;

```

}

OUTPUT:



The screenshot displays the Programiz Online C Compiler interface. The left pane shows the source code for a C program named `main.c`. The code counts characters, words, and lines in an input string. The right pane shows the output of the program, which is successful and displays the counts for characters, words, and lines.

```
19-  for (int i = 0; input[i] != '\0'; i++) {
20-      char_count++;
21-
22-      if (input[i] == '\n')
23-          line_count++;
24-
25-      if (isspace(input[i])) {
26-          in_word = 0; // End of a word
27-      } else if (!in_word) {
28-          in_word = 1;
29-          word_count++; // Start of a new word
30-      }
31-  }
32-
33-  // Print results
34-  printf("Output:\n");
35-  printf("Characters: %d\n", char_count);
36-  printf("Words: %d\n", word_count);
37-  printf("Lines: %d\n", line_count);
38-
39-  return 0;
40- }
```

Output:

```
Input Source Program:
#include <stdio.h>
int main() {
    int a = 10, b = 20;
    printf("Hello, World!");
    return 0;
}

Output:
Characters: 101
Words: 17
Lines: 7

=== Code Execution Successful ===
```